# SQL Anywhere® サーバー
# データベース管理

バージョン 12.0.1

2012 年 1 月

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

バージョン 12.0.1
2012 年 1 月

Copyright © 2012 iAnywhere Solutions, Inc. Portions copyright © 2012 Sybase, Inc. All rights reserved.

iAnywhere との間に書面による合意がないかぎり、このマニュアルは現状のまま提供されるものであり、その使用または記載内容の誤りに対して一切の責任を負いません。

次の条件に従うかぎり、このマニュアルの一部または全体を使用、印刷、複製、配布することができます。1) マニュアルの一部または全体にかかわらず、ここに示したものとそれ以外のすべての版権と商標の表示をすべてのコピーに含めること。2) マニュアルに変更を加えないこと。3) iAnywhere 以外の人間がマニュアルの著者または情報源であるかのように示す一切の行為をしないこと。

iAnywhere®、Sybase®、http://www.sybase.com/detail?id=1011207 に示す商標は Sybase, Inc. またはその関連会社の商標です。® は米国での登録商標を示します。

このマニュアルに記載されているその他の会社名と製品名は各社の商標である場合があります。

# 目次

はじめに ..... v

データベースの開始とデータベースへの接続 ..... 1

チュートリアル：サンプルデータベースへの接続 ..... 1
SQL Anywhere データベース接続 ..... 5
通信プロトコル ..... 96
チュートリアル：SQL Anywhere データベースの作成 ..... 108
データベースファイルのタイプ ..... 112
SQL Anywhere データベースサーバーの使用 ..... 141
SQL Anywhere データベースサーバーの構文 ..... 189
接続パラメーター ..... 320
ネットワークプロトコルオプション ..... 369
SQL Anywhere for Windows Mobile ..... 403

データベース設定 ..... 443

SQL Anywhere の環境変数 ..... 443
ファイルロケーションとインストール設定 ..... 458
国際言語と文字セット ..... 467
ユーザー ID、権限、パーミッション ..... 512
データベースオプション ..... 568
接続、データベース、データベースサーバーのプロパティ ..... 713
SQL Anywhere のサイズと数の制限 ..... 811

データベース管理 ..... 815

SQL Anywhere グラフィカル管理ツール ..... 815
データベース管理ユーティリティ ..... 909

データベースのメンテナンス ..... 1071

バックアップとデータリカバリ ..... 1071
データベースの検証 ..... 1117

スケジュールとイベントの使用によるタスクの自動化 ..... 1123
SQL Anywhere の高可用性 ..... 1136
SQL Anywhere の読み込み専用のスケールアウト ..... 1188
SQL Anywhere データベースのトラブルシューティング ..... 1209

データベースのモニタリング ..... 1217

SQL Anywhere モニター ..... 1217
SQL Anywhere SNMP Extension Agent ..... 1319

セキュリティ ..... 1379

データのセキュリティ ..... 1379
トランスポートレイヤーセキュリティ ..... 1410

レプリケーション ..... 1435

Open Server としての SQL Anywhere ..... 1435

索引 ..... 1447

# はじめに

このマニュアルでは、SQL Anywhere データベースを実行、管理、構成する方法について説明します。データベース接続、データベースサーバー、データベースファイル、バックアッププロシージャー、セキュリティ、高可用性、管理ユーティリティとオプションについて説明します。

# データベースの開始とデータベースへの接続

この項では、SQL Anywhere データベースサーバーの起動方法と、クライアントアプリケーションからデータベースに接続する方法について説明します。

## チュートリアル：サンプルデータベースへの接続

SQL Anywhere サンプルデータベースは、限られた種類のスポーツ衣料品を製造する小企業を想定して作られています。データベースには、この企業の内部情報 (従業員、部署、経理データ)、製品情報 (製品)、販売情報 (受注、顧客、連絡先) が入っています。サンプルデータベースに入っている情報は、すべて架空のものです。

**参照**

- 「SQL Anywhere サンプルデータベース」『SQL Anywhere 12 紹介』

## レッスン 1：SQL Anywhere データベースサーバーの起動とデータベースサーバーメッセージウィンドウの表示

**♦ サンプルデータベースを実行するパーソナルデータベースサーバーを起動するには、次の手順に従います (コマンドラインの場合)。**

1. 次のコマンドを入力して、パーソナルデータベースサーバーを起動し、-n サーバーオプションを使用してデータベースサーバー名を demo12 に設定し、サンプルデータベースに接続します。

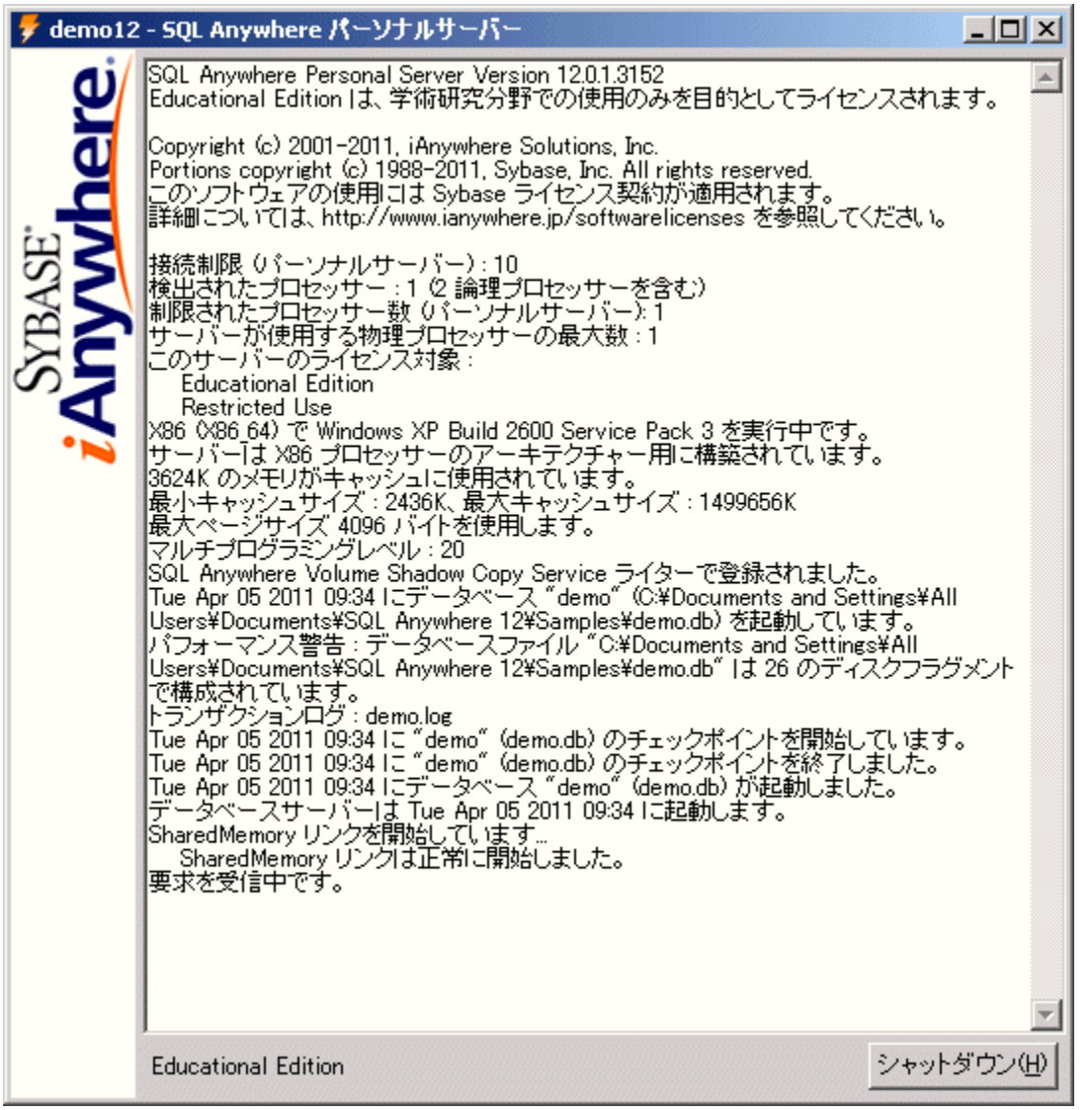
   ```
   dbeng12 -n demo12 "%SQLANYSAMP12%¥demo.db"
   ```

   Windows の場合、データベースサーバーはシステムトレイにアイコンとして表示されます。

   サンプルデータベースを実行するパーソナルデータベースサーバーが正常に起動されました。ただし、データベース内のデータを表示したり操作したりすることはまだできません。

   SQL Anywhere パーソナルサーバーのアイコンが単に表示されているだけです。

2. Windows では、システムトレイで SQL Anywhere パーソナルサーバーアイコンをダブルクリックして、データベースサーバーメッセージウィンドウを表示します。




データベースサーバーメッセージウィンドウには、次の情報が表示されます。

- **サーバー名**　タイトルバーに表示されている名前 (この例では demo12) が「サーバー名」です。このチュートリアルでは、サーバー名を -n サーバーオプションを使用して割り当てています。サーバー名を指定しない場合は、最初に起動されたデータベースの名前になります。この名前は、アプリケーションがデータベースに接続するときに使用します。
- **バージョンとビルド番号**　サーバー名に続く数字 (**12.0.1.3600** など) は、バージョン番号とビルド番号です。バージョン番号は SQL Anywhere の特定のリリースを表し、ビルド番号はコンパイル済みソフトウェアの特定のインスタンスを表します。
- **起動情報**　データベースサーバーは、起動時に、データベース要求を処理するときに使用するメモリを別に設定します。これを「キャッシュ」と呼びます。キャッシュメモリの量

は、このウィンドウに表示されます。キャッシュは固定サイズの「ページ」で構成されていますが、このページのサイズもウィンドウ内に表示されます。

- **データベース情報**　データベースファイルの名前とそのトランザクションログファイルがウィンドウに表示されます。

  この例では、起動時のキャッシュサイズとページのサイズはデフォルト値になっています。このチュートリアルの場合も含め、多くの場合デフォルトの起動オプションが適しています。

3. 「レッスン 2：データベースへの接続」3 ページに進みます。

**参照**


- 「サンプルデータベースへの接続 (Sybase Central の場合)」5 ページ
- 「データベースサーバーの起動」144 ページ
- 「SQL Anywhere データベースサーバーの使用」141 ページ
- 「SQL Anywhere データベースサーバーの構文」189 ページ


# レッスン 2：データベースへの接続

このレッスンでは、サンプルデータベースに接続します。Sybase Central または Interactive SQL を使用して接続できます。

**♦ サンプルデータベースへの接続**

1. Sybase Central または Interactive SQL を起動します。

   - Sybase Central を起動するには、**[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[Sybase Central]** をクリックします。
   - Interactive SQL を起動するには、**[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[Interactive SQL]** をクリックします。

2. **[接続]** » **[SQL Anywhere 12 に接続]** をクリックします。

3. データベースに接続するための情報を以下のフィールドに入力します。

   a. **[ユーザー ID]** フィールドに **DBA** と入力します。

   b. **[パスワード]** フィールドに **sql** と入力します。

   c. **[アクション]** ドロップダウンリストから、**[このコンピューターで稼働しているデータベースに接続]** を選択します。

   d. **[サーバー名]** フィールドに **demo12** と入力します。

   > **注意**
   > コンピューターは、組み込みの SQL Anywhere サーバーを含むソフトウェアを実行している可能性があり、その場合、気づかずに複数のサーバーが実行されているかもしれません。常にサーバー名を指定する (ServerName=*server-name*) ことをおすすめします。

4. **[接続]** をクリックします。

管理ツールがデータベースに接続されます。

5. 省略可能です。Interactive SQL の Customers テーブルを問い合わせます。
   a. **[SQL 文]** ウィンドウ枠で、次のコマンドを入力します。

      ```
      SELECT * FROM Customers;
      ```

   b. 文を実行します。[F5] キーを押すか、**[SQL]** » **[実行]** を選択して、文を実行します。**[結果]** ウィンドウ枠に結果セットが表示されます。

6. 「レッスン 3：データベースサーバーの停止」4 ページに進みます。

**参照**

- 「サンプルデータベースへの接続 (Sybase Central の場合)」5 ページ

## レッスン 3：データベースサーバーの停止

「レッスン 1：SQL Anywhere データベースサーバーの起動とデータベースサーバーメッセージウィンドウの表示」 で起動したデータベースサーバーを停止できます。

Windows では、データベースサーバーメッセージウィンドウで [シャットダウン] をクリックしてデータベースサーバーを停止できます。

**♦ サンプルデータベースを実行しているデータベースサーバーを停止するには、次の手順に従います (Windows の場合)。**

1. システムトレイにある SQL Anywhere のアイコンをダブルクリックします。
2. **[シャットダウン]** クリックします。
3. データベースに接続があることを促すメッセージが表示されたら、**[はい]** をクリックします。

**♦ サンプルデータベースを実行するデータベースサーバーを停止するには、次の手順に従います (コマンドラインの場合)。**

1. 次のコマンドを実行して、サンプルデータベースを実行するパーソナルデータベースサーバーを停止します。

   ```
   dbstop demo12
   ```

   サーバー停止ユーティリティ (dbstop) はコマンドプロンプトでのみ実行できます。

2. データベースに接続があることを促すメッセージが表示されたら、**Y** を入力します。

「レッスン 4：(オプション) サンプルデータベースのリストア」5 ページに進みます。

**参照**

- 「サーバー停止ユーティリティ (dbstop)」1024 ページ
- 「SQL Anywhere データベースサーバーの使用」141 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## レッスン 4 : (オプション) サンプルデータベースのリストア

サンプルデータベースに変更を加えた場合、次の手順を使用すると、サンプルデータベースをリストアできます。

**♦ サンプルデータベースをリストアするには、次の手順に従います。**

● 次のコマンドを実行して、データを含むサンプルデータベースを再作成します。

```
newdemo "%SQLANYSAMP12%¥demo.db"
```

**参照**

● 「サンプルデータベースの再作成 (demo.db)」『SQL Anywhere 12 紹介』

# SQL Anywhere データベース接続

## サンプルデータベースへの接続 (Sybase Central の場合)

このマニュアルでは、多くの例で SQL Anywhere のサンプルデータベースが使用されています。このデータベースに接続し、SQL Anywhere 機能について探索します。

**前提条件**

なし

**内容と備考**

次のとおりです。

**♦ サンプルデータベースに接続するには、次の手順に従います (Sybase Central または Interactive SQL)。**

1. Sybase Central または Interactive SQL を起動してから、**[接続]** ウィンドウを開きます。
   a. **[接続]** ウィンドウが表示されない場合は、次のいずれかを行います。
      ●Sybase Central で、**[接続]** » **[SQL Anywhere 12 に接続]** をクリックします。
      ●Interactive SQL で、**[SQL]** » **[接続]** をクリックします。
   b. Interactive SQL を使用しており、データベースタイプが SQL Anywhere ではない場合、**[データベースタイプを変更]** をクリックしてから **[SQL Anywhere]** をクリックします。
2. **[アクション]** ドロップダウンリストで、**[ODBC データソースを使用した接続]** をクリックします。
3. **[ODBC データソース名]** をクリックし、その下のフィールドに **SQL Anywhere 12 Demo** と入力します。

**注意**

- この接続では、ユーザー ID とパスワードはすでに SQL Anywhere 12 Demo データソースにあるので入力する必要はありません。

  運用環境ではパスワードを ODBC データソースに格納しないことをおすすめします。「ODBC データソースにパスワードを含めない」1383 ページを参照してください。

- データベースサーバーは接続の前に実行中である必要はありません。その理由は、SQL Anywhere 12 Demo データソースがデータベースサーバーを自動的に開始するのに十分な情報を含んでいるからです。「SQL Anywhere サンプルデータベース」『SQL Anywhere 12 紹介』を参照してください。

4. オプション。このウィンドウで作成される接続文字列をコピーするには、**[ツール]** » **[接続文字列をクリップボードにコピー]** をクリックします。

5. **[接続]** をクリックします。

**結果**

データベースサーバーとデータベースが起動され、Interactive SQL または Sybase Central はデータベースに接続します。

**次の手順**

なし。

**参照**

- 「SQL Anywhere サンプルデータベース」『SQL Anywhere 12 紹介』
- 「ODBC データソースを使用した接続 (コマンドラインの場合)」18 ページ

# サンプルデータベースへの接続 (Interactive SQL の場合)

このマニュアルでは、多くの例で SQL Anywhere のサンプルデータベースが使用されています。このデータベースに接続し、SQL Anywhere について探索します。データベースに接続するには、コマンドプロンプトを使用して Interactive SQL を起動します。

**前提条件**

なし。

**内容と備考**

次のとおりです。

**♦ Interactive SQL からサンプルデータベースに接続するには、次の手順に従います (接続文字列を指定したコマンドラインの場合)。**

- 次のコマンドを実行して、ODBC データソースを使ってサンプルデータベースに接続します。

```
dbisql -c "DSN=SQL Anywhere 12 Demo"
```

### 結果

Interactive SQL が起動され、サンプルデータベースに接続します。

### 次の手順

なし。

### 参照

- 「SQL Anywhere サンプルデータベース」『SQL Anywhere 12 紹介』

## UNIX でのサンプルデータベースへの接続

このマニュアルでは、多くの例で SQL Anywhere のサンプルデータベースが使用されています。このデータベースに接続し、SQL Anywhere について探索します。

### 前提条件

次の手順では、Interactive SQL を使用してデータベースに接続します。Java Runtime Environment (JRE) がインストールされていることを確認してください。

一般的な UNIX インストールでは、サンプルデータベースを含めた SQL Anywhere ソフトウェアは、書き込みパーミッションのないディレクトリにインストールされます。まず、ターミナルウィンドウで書き込み可能なフォルダーに移動します。次のコマンドを実行して、サンプルデータベースをフォルダーにコピーします。

```
cp "samples-dir/demo.db".
```

データベースサーバーでサンプルデータベースを起動します。

```
dbeng12 demo.db
```

### 内容と備考

Mac OS X を使用して接続するには、「Mac OS X でのサンプルデータベースへの接続」8 ページを参照してください。

**♦ Interactive SQL からサンプルデータベースに接続するには、次の手順に従います (UNIX の場合)。**

1. ターミナルウィンドウで、Interactive SQL を起動します。

   ```
   dbisql
   ```

   デフォルトで **[接続]** ウィンドウが表示されます。

2. **[接続]** ウィンドウに次の情報が表示されます。
   a. **[ユーザー ID]** フィールドに **DBA** と入力します。

b. **[パスワード]** フィールドに **sql** と入力します。

c. **[アクション]** ドロップダウンリストで、**[このコンピューターで稼働しているデータベースに接続]** をクリックします。

d. **[サーバー名]** フィールドに **demo** と入力します。

e. オプション。このウィンドウで作成される接続文字列をコピーするには、**[ツール]** » **[接続文字列をクリップボードにコピー]** をクリックします。

f. **[接続]** をクリックします。

### 結果

Interactive SQL が起動され、サンプルデータベースに接続します。

### 次の手順

なし。

### 参照

- 「SQL Anywhere サンプルデータベース」『SQL Anywhere 12 紹介』
- 「UNIX での ODBC データソース」48 ページ

## Mac OS X でのサンプルデータベースへの接続

このマニュアルでは、多くの例で SQL Anywhere のサンプルデータベースが使用されています。このデータベースに接続し、SQL Anywhere について探索します。

### 前提条件

- **Finder** で、SQL Anywhere サンプルデータベースを検索してファイルを読み込み／書き込みアクセス権がある場所にコピーします。デフォルトでは、サンプルデータベースは */Applications/SQLAnywhere12/demo.db* にあります。

- データベースサーバーを起動します。**Finder** で、**DBLauncher** をダブルクリックします。デフォルトでは、**DBLauncher** は */Applications/SQLAnywhere12* にあります。

  1. **[Database]** フィールドで、SQL Anywhere サンプルデータベースのロケーションを参照します (*/Applications/SQLAnywhere12/System/demo.db* など)。

  2. **[Server Name]** フィールドに **demo** と入力します。

  3. **[Local Server]** をクリックします。

     **[Local Server]** オプションを選択すると、ネットワーク経由でクライアント／サーバー通信を行えなくなります。

  4. **[Start]** をクリックして、**demo** という名前のパーソナルデータベースサーバーを起動します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 内容と備考

次のとおりです。

**♦ Interactive SQL からサンプルデータベースに接続するには、次の手順に従います (Mac OS X の場合)。**

1. **Finder** で、*/Applications/SQLAnywhere12* にある **[Interactive SQL]** をダブルクリックします。
2. **[接続]** ウィンドウに次の情報を入力して、SQL Anywhere サンプルデータベースに接続します。
   - **[認証]** ドロップダウンリストで、**[データベース]** をクリックします。
   - **[ユーザー ID]** フィールドに **DBA** と入力します。
   - **[パスワード]** フィールドに **sql** と入力します。
   - **[アクション]** ドロップダウンリストで、**[このコンピューターで稼働しているデータベースに接続]** をクリックし、コンピューター上で実行されているデータベースに接続します。
   - **[サーバー名]** フィールドに **demo** と入力します。
   - **[データベース名]** フィールドに **demo** と入力します。
   - オプション。このウィンドウで作成される接続文字列をコピーするには、**[ツール]** » **[接続文字列をクリップボードにコピー]** をクリックします。
   - **[接続]** をクリックします。

### 結果

Interactive SQL が起動され、サンプルデータベースに接続します。

### 次の手順

なし。

### 参照


- 「SQL Anywhere サンプルデータベース」『SQL Anywhere 12 紹介』
- 「UNIX でのサンプルデータベースへの接続」7 ページ
- 「ODBC データソースの作成 (Mac OS X の場合)」43 ページ


## Sybase Central の接続プロファイル

データベースサーバーまたはデータベースに最初に接続するときには、ユーザー ID やパスワードなどの接続パラメーターを指定します。この情報は、その後の接続時に再び入力する必要があります。Sybase Central を使用するときの時間の節約と接続プロセスの簡略化のため、接続プロファイルを作成して、各データベースの接続パラメーターを保存できます。

接続プロファイルを使用したり管理したりするには、**[接続]** » **[接続プロファイル]** をクリックして、**[接続プロファイル]** ウィンドウを開きます。**[接続プロファイル]** ウィンドウを使用して、次の操作を実行します。

- 新しい接続プロファイルの作成
- Sybase Central の起動時に自動的に接続するプロファイルの設定
- 既存のプロファイルの編集
- 接続プロファイルを使用した接続
- プロファイルの削除
- 接続プロファイルのインポートとエクスポート

**注意**
接続プロファイルは Sybase Central 固有です。ODBC アプリケーションを構築している場合は、ODBC データソースを使用して接続プロファイルに似た機能を実現できます。「ODBC データソース」36 ページを参照してください。

## 接続プロファイルの作成 (Sybase Central の場合)

Sybase Central での時間の節約と接続プロセスの簡略化のため、接続プロファイルを作成して、各データベースの接続パラメーターを保存します。

### 前提条件

なし。

### 内容と備考

**警告**
ユーザー ID、パスワード (暗号化の有無は不問)、データベースキーを接続プロファイルに保存することはおすすめしません。

接続プロファイルは Sybase Central 固有です。ODBC アプリケーションを構築している場合は、ODBC データソースを使用して接続プロファイルに似た機能を実現できます。「ODBC データソース」36 ページを参照してください。

**♦ 新しい接続プロファイルを作成するには、次の手順に従います (Sybase Central の場合)。**

1. **[接続]** » **[接続プロファイル]** をクリックします。
2. **[新規]** をクリックします。
3. **[名前]** フィールドに新しいプロファイルの名前を入力します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

4. **[新しい接続プロファイル]** をクリックし、リストから適切なプラグインを選択します。プラグインは、**SQL Anywhere 12** や **Ultra Light 12** などの製品です。

   既存のプロファイルに基づいて新しい接続プロファイルを作成する場合は、**[接続プロファイルをコピー]** をクリックし、**[既存の接続プロファイル]** リストからプロファイルを選択します。

5. 他のユーザーがプロファイルにアクセスできるようにするには、**[この接続プロファイルを他のユーザーと共有する]** をクリックします。この設定は UNIX などのマルチユーザープラットフォームの場合に便利です。

6. **[OK]** をクリックします。

7. **[接続プロファイルの編集]** ウィンドウで、必要な値を入力し、**[保存]** をクリックしてウィンドウを閉じます。

### 結果

接続プロファイルが作成され、それを使用してデータベースに接続することができます。

### 次の手順

なし。

### 参照

- 「接続プロファイルを使用した自動的な接続 (Sybase Central の場合)」11 ページ
- 「接続プロファイルのインポート (Sybase Central の場合)」13 ページ
- 「接続プロファイルのエクスポート (Sybase Central の場合)」13 ページ

## 接続プロファイルを使用した自動的な接続 (Sybase Central の場合)

Sybase Central と同じ接続プロファイルを常用する場合は、デフォルトでその接続プロファイルを使用するような Sybase Central の設定を検討してください。Sybase Central を起動すると、接続プロファイルを使用してデータベースに自動的に接続します。

### 前提条件

接続プロファイルを定義する必要があります。

### 内容と備考

次のとおりです。

**♦ 接続プロファイルを使用した自動的な接続 (Sybase Central の場合)**

1. **[接続]** » **[接続プロファイル]** をクリックします。

2. **[接続プロファイル]** リストで接続プロファイルを選択します。

3. **[スタートアップの設定]** をクリックし、**[起動時に使用]** の設定を **[いいえ]** から **[はい]** に変更します。

**結果**

Sybase Central を起動すると、接続プロファイルを使用してデータベースに自動的に接続します。

**次の手順**

なし。

**参照**


- 「接続プロファイルのインポート (Sybase Central の場合)」 13 ページ
- 「接続プロファイルのエクスポート (Sybase Central の場合)」 13 ページ
- 「接続プロファイルの作成 (Sybase Central の場合)」 10 ページ


## 接続プロファイルの編集 (Sybase Central の場合)

接続プロファイルの接続情報が変更された場合は、接続プロファイルを編集して接続パラメーターを保存します。

**前提条件**

接続プロファイルを定義する必要があります。

**内容と備考**

次のとおりです。

**♦ 接続プロファイルを編集するには、次の手順に従います (Sybase Central の場合)。**

1. **[接続]** » **[接続プロファイル]** をクリックします。
2. **[接続プロファイル]** リストで接続プロファイルを選択します。
3. **[編集]** をクリックします。
4. **[接続プロファイルの編集]** ウィンドウで値を編集します。
5. **[保存]** をクリックします。

**結果**

接続プロファイルが保存されます。

**次の手順**

なし。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**参照**


- 「接続プロファイルを使用した自動的な接続 (Sybase Central の場合)」 11 ページ
- 「接続プロファイルのインポート (Sybase Central の場合)」 13 ページ
- 「接続プロファイルのエクスポート (Sybase Central の場合)」 13 ページ
- 「接続プロファイルの作成 (Sybase Central の場合)」 10 ページ


## 接続プロファイルのインポート (Sybase Central の場合)

他の場所で作成された接続プロファイルを使用するには、その接続プロファイルをインポートします。

### 前提条件

エクスポートされた接続プロファイルを含むファイルを定義する必要があります。

### 内容と備考

次のとおりです。

**♦ 接続プロファイルをインポートするには、次の手順に従います (Sybase Central の場合)。**

1. **[接続]** » **[接続プロファイル]** をクリックします。
2. **[インポート]** をクリックします。
3. **[ファイル名]** フィールドに、インポートする接続プロファイルファイルの名前を入力します。
4. **[OK]** をクリックします。

### 結果

接続プロファイルがインポートされます。

### 次の手順

なし。


- 「接続プロファイルのエクスポート (Sybase Central の場合)」 13 ページ
- 「接続プロファイルを使用した自動的な接続 (Sybase Central の場合)」 11 ページ
- 「接続プロファイルの作成 (Sybase Central の場合)」 10 ページ


## 接続プロファイルのエクスポート (Sybase Central の場合)

他のコンピューターのユーザーと接続プロファイルを共有する、または接続プロファイルをバックアップするには、接続プロファイルをファイルにエクスポートします。

### 前提条件

接続プロファイルを定義する必要があります。

### 内容と備考

次のとおりです。

**♦ 接続プロファイルをエクスポートするには、次の手順に従います (Sybase Central の場合)。**

1. **[接続]** » **[接続プロファイル]** をクリックします。
2. **[接続プロファイル]** リストで接続プロファイルを選択します。
3. **[エクスポート]** をクリックします。
4. **[ファイル名]** フィールドに接続プロファイルのファイル名を入力します。
5. **[保存]** をクリックします。

### 結果

接続プロファイルは、指定されたファイルにエクスポートされます。

### 次の手順

接続プロファイルをインポートします。


- 「接続プロファイルのインポート (Sybase Central の場合)」 13 ページ
- 「接続プロファイルを使用した自動的な接続 (Sybase Central の場合)」 11 ページ
- 「接続プロファイルの作成 (Sybase Central の場合)」 10 ページ


## [接続] ウィンドウを開く (Sybase Central の場合)

Sybase Central、Interactive SQL、SQL Anywhere コンソールユーティリティからサーバーまたはデータベースに接続するには、**[接続]** ウィンドウを使用して接続パラメーターを指定します。

### 前提条件

なし

### 内容と備考

デフォルトでは、管理ツールを起動すると **[接続]** ウィンドウが開きます。**[接続]** ウィンドウが自動的に開かない場合は、次のいずれかの手順で開きます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**♦ [接続] ウィンドウを開くには、次の手順に従います (Sybase Central の場合)。**

1. Sybase Central を起動します。「Sybase Central の起動 (Windows の場合)」816 ページを参照してください。

2. **[接続]** » **[SQL Anywhere 12 に接続]** をクリックします。

   [F11] キーを押して **[接続]** メニューを開くこともできます。

3. データベースに接続します。 「SQL Anywhere データベース接続」30 ページを参照してください。

**♦ [接続] ウィンドウを開くには、次の手順に従います (Interactive SQL の場合)。**

1. Interactive SQL を起動します。「Interactive SQL の起動」841 ページを参照してください。

2. **[SQL]** » **[接続]** をクリックします。

   [F11] キーを押して **[接続]** ウィンドウを開くこともできます。

3. データベースに接続します。「SQL Anywhere データベース接続」30 ページを参照してください。

**♦ [接続] ウィンドウを開くには、次の手順に従います (SQL Anywhere コンソールユーティリティの場合)。**

1. SQL Anywhere コンソールユーティリティを起動します。「SQL Anywhere コンソールユーティリティ (dbconsole) の起動」905 ページを参照してください。

2. **[コマンド]** » **[接続]** をクリックします。

3. データベースに接続します。「SQL Anywhere データベース接続」30 ページを参照してください。

### 結果

**[接続]** ウィンドウが開きます。

### 次の手順

**[接続]** ウィンドウに入力する情報は、管理ツール実行中は保持されます。

## 参照


- 「SQL Anywhere データベース接続」30 ページ
- 「SQL Anywhere コンソールユーティリティ (dbconsole) の起動」905 ページ

# このコンピューターのデータベースを起動して接続する (dbeng12)

**[このコンピューターのデータベースを起動して接続]** オプションを使用して **[接続]** ウィンドウを使用して、サーバーとデータベースを起動して、すべてを同じコンピューターからこのローカルデータベースに接続することができます。

### 前提条件

なし。

### 内容と備考

次のとおりです。

**♦ データベースを起動して接続するには、次の手順に従います (コマンドラインの場合)。**

● 次のコマンドを実行して Interactive SQL を起動し、サンプルデータベースを起動して接続します。

```
dbisql -c "UID=DBA;PWD=sql;ServerName=Demo12;DBF=%SQLANYSAMP12%¥demo.db"
```

**♦ このコンピューターのデータベースを起動して接続する (dbeng12)**

1. Sybase Central または Interactive SQL で **[接続]** ウィンドウを開きます。「[接続] ウィンドウを開く (Sybase Central の場合)」14 ページを参照してください。
2. データベースに接続するための情報を以下のフィールドに入力します。
   a. **[認証]** ドロップダウンリストで、**[データベース]** をクリックします。
   b. **[ユーザー ID]** フィールドにユーザー名を入力します。たとえば、**DBA** と入力します。
   c. **[パスワード]** フィールドにユーザーのパスワードを入力します。たとえば、**sql** と入力します。
   d. **[アクション]** ドロップダウンリストで、**[このコンピューターのデータベースを起動して接続]** をクリックします。
   e. **[データベースファイル]** フィールドで、データベースファイルパス、ファイル名、ファイル拡張子を指定します。たとえば、*C:¥Documents and Settings¥All Users¥Documents¥SQL Anywhere 12¥Samples¥demo.db* のように入力します。
   f. 次回以降の接続のために、ファイル名とは異なるデータベース名を作成する場合は、**[データベース名]** フィールドに名前を入力します。ファイルのパスや拡張子は入力しないでください。
   g. **[サーバー名]** フィールドにサーバーの名前を指定します。たとえば、**demo12** と入力します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

> **注意**
> コンピューターは、組み込みの SQL Anywhere サーバーを含むソフトウェアを実行している可能性があり、その場合、気づかずに複数のサーバーが実行されているかもしれません。常にサーバー名を指定する (ServerName=*server-name*) ことをおすすめします。

3. オプション。このウィンドウで作成される接続文字列をコピーするには、**[ツール]** » **[接続文字列をクリップボードにコピー]** をクリックします。

4. **[接続]** をクリックします。

### 結果

データベースサーバーとデータベースが起動して、管理ツールはデータベースに接続します。

### 次の手順

なし。

### 参照

- 「チュートリアル：サンプルデータベースへの接続」1 ページ
- 「[接続] ウィンドウを開く (Sybase Central の場合)」14 ページ

## このコンピューターで稼働しているデータベースに接続 (dbeng 12)

**[このコンピューターで稼働しているデータベースに接続]** オプションを指定して、**[接続]** を使用することで、現在稼働しているローカルデータベースに接続することができます。

### 前提条件

この手順は、使用しているコンピューター上のデータベースサーバーでデータベースが稼働していることを前提としています。まだ稼働していない場合には、データベースサーバーとデータベースを起動します。たとえば、次のコマンドを実行して、パーソナルサーバー上で SQL Anywhere 12 サンプルデータベースを起動します。

```
dbeng12 -n demo12 "%SQLANYSAMP12%¥demo.db"
```

### 内容と備考

次のとおりです。

#### ♦ このコンピューターで稼働しているデータベースに接続する (dbeng12)

- 次のコマンドを実行して、Interactive SQL を起動し、データベースサーバーに接続します。

```
dbisql -c "UID=DBA;PWD=sql;ServerName=demo12"
```

♦ **このコンピューターで稼働しているデータベースに接続する (dbeng12)**

1. **[接続]** ウィンドウを開きます。 「[接続] ウィンドウを開く (Sybase Central の場合)」14 ページ を参照してください。

2. データベースに接続するための情報を以下のフィールドに入力します。
   a. **[認証]** ドロップダウンリストで **[データベース]** をクリックします。
   b. **[ユーザー ID]** フィールドにユーザー名を入力します。たとえば、**DBA** と入力します。
   c. **[パスワード]** フィールドにユーザーのパスワードを入力します。たとえば、**sql** と入力します。
   d. **[アクション]** ドロップダウンリストで、**[このコンピューターで稼働しているデータベースに接続]** をクリックし、コンピューター上で実行されているデータベースに接続します。
   e. **[サーバー名]** フィールドに、実行中のデータベースサーバーの名前を入力します。たとえば、**demo12** と入力します。

   > **注意**
   > コンピューターは、組み込みの SQL Anywhere サーバーを含むソフトウェアを実行している可能性があり、その場合、気づかずに複数のサーバーが実行されているかもしれません。常にサーバー名を指定する (ServerName=*server-name*) ことをおすすめします。

   f. **[データベース名]** フィールドにデータベースの名前を入力します。たとえば、**demo** と入力します。

3. オプション。このウィンドウで作成される接続文字列をコピーするには、**[ツール]** » **[接続文字列をクリップボードにコピー]** をクリックします。

4. **[接続]** をクリックします。

**結果**

アプリケーションがデータベースに接続します。

**次の手順**

なし。

**参照**

- 「チュートリアル：サンプルデータベースへの接続」1 ページ
- 「[接続] ウィンドウを開く (Sybase Central の場合)」14 ページ

# ODBC データソースを使用した接続 (コマンドラインの場合)

ODBC データソースを使用してデータベースに接続するには、DataSourceName (DSN) 接続パラメーターを使用します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 前提条件

接続するデータベースの ODBC データソースを使用します。

SQL Anywhere 12 Demo データソースは、データベースサーバーとデータベースがまだ稼働していない場合に、データベースサーバーとデータベースを起動するように設定されます。ODBC データソースがデータベースとデータベースサーバーを起動するように設定されていない場合は、ユーザーがこれらを起動する必要があります。たとえば、次のコマンドを実行して、SQL Anywhere 12 サンプルデータベースを起動します。

```
dbeng12 -n demo12 "%SQLANYSAMP12%¥demo.db"
```

### 内容と備考

**注意**
接続文字列と ODBC データソースがどちらも同じ接続パラメーターを指定する場合、接続文字列の値が使用され、データソースからの値は無視されます。「接続パラメーターの構文ルール」34 ページを参照してください。

### ♦ ODBC データソースを作成するには、次の手順に従います (コマンドラインの場合)。

- ODBC データソース名を使って接続するために次のコマンドを実行します。

  ```
  dbisql -c "DSN=SQL Anywhere 12 Demo"
  ```

  Interactive SQL が起動され、サンプルデータベースに接続します。

  **注意**
  この接続では、ユーザー ID とパスワードはすでに SQL Anywhere 12 Demo データソースにあるので入力する必要はありません。

  運用環境ではパスワードを ODBC データソースに格納しないことをおすすめします。「ODBC データソースにパスワードを含めない」1383 ページを参照してください。

### 結果

アプリケーションがサンプルデータベースに接続します。

### 次の手順

なし。

## 参照


- 「DataSourceName (DSN) 接続パラメーター」338 ページ
- 「ODBC データソース」36 ページ
- 「UNIX での ODBC データソース」48 ページ

# ODBC データソースを使用した接続 (Sybase Central または Interactive SQL の場合)

ODBC データソースを使用してデータベースに接続するには、**[ODBC データソース名]** オプションを指定して **[接続]** ウィンドウを使用します。

### 前提条件

接続するデータベースの ODBC データソースを使用します。

### 内容と備考

SQL Anywhere 12 Demo データソースは、データベースサーバーとデータベースがまだ稼働していない場合、データベースサーバーとデータベースを起動するように設定されています。ODBC データソースがデータベースとデータベースサーバーを起動するように設定されていない場合は、ユーザーがこれらを起動する必要があります。たとえば、次のコマンドを実行して、SQL Anywhere 12 サンプルデータベースを起動します。

```
dbeng12 -n demo12 "%SQLANYSAMP12%¥demo.db"
```

**♦ ODBC データソースを使用して接続するには、次の手順に従います (Sybase Central または Interactive SQL の場合)。**

1. Sybase Central または Interactive SQL から **[接続]** ウィンドウを開きます。「[接続] ウィンドウを開く (Sybase Central の場合)」14 ページを参照してください。
2. データベースに接続するための情報を以下のフィールドに入力します。
   a. **[認証]** ドロップダウンリストで **[データベース]** をクリックします。
   b. **[ユーザー ID]** フィールドにユーザー名を入力します。たとえば、**DBA** と入力します。
   c. **[パスワード]** フィールドにユーザーのパスワードを入力します。たとえば、**sql** と入力します。
   d. **[アクション]** ドロップダウンリストで、**[ODBC データソースを使用した接続]** をクリックします。
   e. 次のいずれかを行います。
      - **[ODBC データソース名]** をクリックし、データソースの名前を入力します。たとえば、**SQL Anywhere 12 Demo** と入力します。
      - **[ODBC データソースファイル]** をクリックし、ファイルに格納されているデータソースを参照する FileDataSourceName (FILEDSN) 接続パラメーターを入力します。
3. オプション。このウィンドウで作成される接続文字列をコピーするには、**[ツール]** » **[接続文字列をクリップボードにコピー]** をクリックします。
4. **[接続]** をクリックします。

### 結果

管理ツールがデータベースに接続されます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 次の手順

なし。

### 参照


- 「ODBC データソース」36 ページ
- 「UNIX での ODBC データソース」48 ページ


## 接続文字列を使用したデータソースによる接続 (Sybase Central または Interactive SQL の場合)

**[接続]** ウィンドウで接続文字列を指定するには、**[接続文字列を使用して接続]** オプションを選択します。ODBC データソースを使用し、接続時に追加のまたは異なるパラメーターを指定する場合に、このオプションは便利です。

### 前提条件

この手順における接続文字列は、DSN 接続パラメーターを使用して SQL Anywhere 12 Demo データソースに接続します。このデータソースは、データベースサーバーとデータベースがまだ稼働していない場合に、データベースサーバーとデータベースを起動するように設定されます。ODBC データソースがデータベースとデータベースサーバーを起動するように設定されていない場合は、ユーザーがこれらを起動する必要があります。たとえば、次のコマンドを実行して、サンプルデータベースを起動します。

```
dbeng12 -n SampleServer "%SQLANYSAMP12%¥demo.db"
```

### 内容と備考

接続文字列と ODBC データソースがどちらも同じ接続パラメーターを指定する場合、接続文字列の値が使用され、データソースからの値は無視されます。「接続パラメーターの構文ルール」34 ページを参照してください。

**♦ 接続文字列を使用してデータソースを使用して接続するには、次の手順に従います (Sybase Central または Interactive SQL の場合)。**

1. Sybase Central または Interactive SQL で、**[接続]** ウィンドウを開きます。 「[接続] ウィンドウを開く (Sybase Central の場合)」14 ページを参照してください。
2. データベースに接続するための情報を以下のフィールドに入力します。
   a. **[認証]** ドロップダウンリストで、**[データベース]** をクリックします。
   b. **[ユーザー ID]** フィールドにユーザー名を入力します。
   c. **[パスワード]** フィールドにユーザーのパスワードを入力します。
   d. **[アクション]** ドロップダウンリストで、**[接続文字列を使用して接続]** をクリックします。
   e. **[パラメーター]** フィールドで、「パラメーター=値」のペアをセミコロンで区切ったリストの形式で、接続パラメーターを入力します。次に例を示します。

```
-c DSN=SQL Anywhere 12 Demo;ServerName=SampleServer
```

データベースサーバーとデータベースが起動して、Interactive SQL はデータベースに接続します。データベースサーバーの名前は SampleServer です。

> **注意**
> コンピューターは、組み込みの SQL Anywhere サーバーを含むソフトウェアを実行している可能性があり、その場合、気づかずに複数のサーバーが実行されているかもしれません。常にサーバー名を指定する (ServerName=*server-name*) ことをおすすめします。

3. **[接続]** をクリックします。

**結果**

管理ツールがデータベースに接続されます。

**次の手順**

なし。

**参照**

- 「接続文字列として渡される接続パラメーター」31 ページ

## 別のコンピューターで稼働しているデータベースに接続 (dbsrv 12) (コマンドラインの場合)

Host 接続パラメーターを使用することで、現在ネットワークデータベースサーバーで稼働しているデータベースに接続することができます。

**前提条件**

別のコンピューターで実行中のデータベースです。

**内容と備考**

次のとおりです。

**♦ 別のコンピューターで実行中のデータベースに接続するには、次の手順に従います (コマンドラインの場合)。**

- 次のコマンドを実行します (my_host を、ネットワークサーバーを実行しているコンピューターのホスト名に置き換えます)。

```
dbisql -c "UID=DBA;PWD=sql;ServerName=demo12;Host=my_host"
```

> **注意**
> コンピューターは、組み込みの SQL Anywhere サーバーを含むソフトウェアを実行している可能性があり、その場合、気づかずに複数のサーバーが実行されているかもしれません。常にサーバー名を指定する (ServerName=*server-name*) ことをおすすめします。

### 結果

アプリケーションがデータベースに接続します。

### 次の手順

なし。

### 参照

- 「Host 接続パラメーター」346 ページ

## 別のコンピューターで実行中のデータベースに接続 (Sybase Central または Interactive SQL の場合)

現在ネットワークデータベースサーバーで稼働しているデータベースへは、**[別のコンピューターで稼働しているデータベースに接続]** オプションを使用し、**[接続]** ウィンドウから接続します。

### 前提条件

この手順は、データベースが別のコンピューターのネットワークサーバー上で実行中であることを前提としています。まだ稼働していない場合には、データベースサーバーとデータベースを起動します。たとえば、次のコマンドを実行して、SQL Anywhere 12 サンプルデータベースを起動します。

```
dbsrv12 -n demo12 "%SQLANYSAMP12%¥demo.db"
```

### 内容と備考

次のとおりです。

**♦ 別のコンピューターで実行中のデータベースに接続するには、次の手順に従います (Sybase Central または Interactive SQL の場合)。**

1. Sybase Central または Interactive SQL で、**[接続]** ウィンドウを開きます。「[接続] ウィンドウを開く (Sybase Central の場合)」14 ページを参照してください。
2. データベースに接続するための情報を以下のフィールドに入力します。
   a. **[認証]** ドロップダウンリストで **[データベース]** をクリックします。
   b. **[ユーザー ID]** フィールドにユーザー名を入力します。たとえば、**DBA** と入力します。
   c. **[パスワード]** フィールドにユーザーのパスワードを入力します。たとえば、**sql** と入力します。

d. **[アクション]** ドロップダウンリストで、**[別のコンピューターで稼働しているデータベースに接続]** をクリックし、別のコンピューター上で実行されているデータベースに接続します。

e. **[ホスト]** フィールドに、データベースサーバーが実行中のコンピューターのホスト名を入力します。

f. **[サーバー名]** フィールドに、サーバー名を入力するか、**[検索]** をクリックします。たとえば、**demo12** と入力します。

> **注意**
> コンピューターは、組み込みの SQL Anywhere サーバーを含むソフトウェアを実行している可能性があり、その場合、気づかずに複数のサーバーが実行されているかもしれません。常にサーバー名を指定する (ServerName=*server-name*) ことをおすすめします。

g. **[データベース名]** フィールドにデータベースの名前を入力します。たとえば、**demo** と入力します。

3. オプション。このウィンドウで作成される接続文字列をコピーするには、**[ツール]** » **[接続文字列をクリップボードにコピー]** をクリックします。

4. **[接続]** をクリックします。

**結果**

管理ツールがデータベースに接続されます。

**次の手順**

なし。

**参照**

- 「チュートリアル：サンプルデータベースへの接続」1 ページ
- 「[接続] ウィンドウを開く (Sybase Central の場合)」14 ページ

# クラウド内で実行しているデータベースへの接続 (dbsrv12)

クラウド内で現在稼働しているデータベースへは、**[接続]** ウィンドウで **[クラウドのコンピューターのデータベースに接続]** オプションを指定し接続します。

**前提条件**

クラウドのデータベースに接続するには、少なくともクラウドのホスト、データベース名、データベースのユーザー ID とパスワードを指定する必要があります。

**内容と備考**

必ずクラウドに接続できるよう、クラウドパートナーとクラウド監視サーバーのホスト名とポート番号を指定することをおすすめします。

♦ **クラウド内で実行中のデータベースに接続 (Sybase Central または Interactive SQL の場合)**

1. Sybase Central または Interactive SQL で、**[接続]** ウィンドウを開きます。
2. 次のフィールドをすべて指定または入力します。
   a. **[認証]** ドロップダウンリストで **[データベース]** をクリックします。
   b. **[ユーザー ID]** フィールドにユーザー名を入力します。たとえば、**DBA** と入力します。
   c. **[パスワード]** フィールドにユーザーのパスワードを入力します。たとえば、**sql** と入力します。
   d. **[アクション]** ドロップダウンリストで、**[クラウドのコンピューターのデータベースに接続]** をクリックし、別のコンピューター上で実行されているデータベースに接続します。
   e. **[ホスト]** フィールドに、クラウド監視サーバーと 1 つのクラウドパートナーのホスト名を入力します。
   f. **[ポート]** フィールドにクラウド監視サーバーを実行している TCP/IP ポートを入力します。その後、クラウドパートナーを実行しているポートを続けます。
   g. **[データベース名]** フィールドにデータベースの名前を入力します。たとえば、**Tenant1** と入力します。
3. オプション。このウィンドウで作成される接続文字列をコピーするには、**[ツール]** » **[接続文字列をクリップボードにコピー]** をクリックします。
4. **[接続]** をクリックします。

**結果**

管理ツールがデータベースに接続されます。

**次の手順**

なし。

## 別のコンピューターのデータベースを起動して接続 (dbsrv12)

おすすめできませんが、別のコンピューター上のデータベースを起動して接続することもできます。

**前提条件**

データベースサーバーがまだ実行されていない場合は、別のコンピューター上でネットワークデータベースサーバーを起動します。たとえば、次のコマンドを実行して、SQL Anywhere 12 Demo データベースを起動します。

```
dbsrv12 -n demo12 -gd all
```

### 内容と備考

**警告**
次のシナリオは一般的ではないので、おすすめしません。これは、ネットワークデータベースサーバーが別のコンピューター上で -gd all オプションを指定されて起動されたことを前提としています。 「-gd dbeng12/dbsrv12 サーバーオプション」233 ページを参照してください。

### ♦ 別のコンピューターのデータベースを起動して接続 (dbsrv12)

● 次のコマンドを実行します (my_host をネットワークサーバーを実行しているコンピューターのホスト名に置き換え、demo_path をデータベースサーバーに相対的なパスに置き換えます)。

```
dbisql -c "UID=DBA;PWD=sql;Server=demo12;DBF=c:¥demo_path¥demo.db;host=my_host"
```

**注意**
コンピューターは、組み込みの SQL Anywhere サーバーを含むソフトウェアを実行している可能性があり、その場合、気づかずに複数のサーバーが実行されているかもしれません。常にサーバー名を指定する (ServerName=*server-name*) ことをおすすめします。

### 結果

アプリケーションがデータベースに接続します。

### 次の手順

なし。

## 別のコンピューターのデータベースを起動して接続 (管理ツールの場合)

おすすめできませんが、**[接続]** ウィンドウを使用して別のコンピューター上のデータベースを起動して接続することもできます。

### 前提条件

データベースサーバーがまだ実行されていない場合は、別のコンピューター上でネットワークデータベースサーバーを起動します。たとえば、次のコマンドを実行して、SQL Anywhere 12 Demo データベースを起動します。

```
dbsrv12 -n demo12 -gd all
```

### 内容と備考

**警告**
次のシナリオは一般的ではないので、おすすめしません。これは、ネットワークデータベースサーバーが別のコンピューター上で -gd all オプションを指定されて起動されたことを前提としています。 「-gd dbeng12/dbsrv12 サーバーオプション」233 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### ♦ 別のコンピューターで実行中のデータベースへの接続 (Sybase Central または Interactive SQL の場合)

1. Sybase Central または Interactive SQL で、**[接続]** ウィンドウを開きます。「[接続] ウィンドウを開く (Sybase Central の場合)」14 ページを参照してください。

2. データベースに接続するための情報を以下のフィールドに入力します。
   a. **[認証]** ドロップダウンリストで **[データベース]** をクリックします。
   b. **[ユーザー ID]** フィールドにユーザー名を入力します。たとえば、**DBA** と入力します。
   c. **[パスワード]** フィールドにユーザーのパスワードを入力します。たとえば、**sql** と入力します。
   d. **[アクション]** ドロップダウンリストで、**[別のコンピューターのデータベースを起動して接続]** をクリックします。
   e. **[データベースファイル]** フィールドで、データベースサーバーに相対的なデータベースファイルのパスと拡張子を入力します。たとえば、**c:¥path_demo¥demo.db** と入力します。
   f. 次回以降の接続のために、ファイル名とは異なるデータベース名を作成する場合は、**[データベース名]** フィールドに名前を入力します。ファイルのパスや拡張子は入力しないでください。たとえば、**demo** と入力します。
   g. **[サーバー名]** フィールドにサーバーの名前を指定します。たとえば、**demo12** と入力します。

   > **注意**
   > コンピューターは、組み込みの SQL Anywhere サーバーを含むソフトウェアを実行している可能性があり、その場合、気づかずに複数のサーバーが実行されているかもしれません。常にサーバー名を指定する (ServerName=*server-name*) ことをおすすめします。

   h. **[ホスト]** フィールドに、データベースサーバーが実行中のコンピューターのホスト名を入力します。

3. オプション。このウィンドウで作成される接続文字列をコピーするには、**[ツール]** » **[接続文字列をクリップボードにコピー]** をクリックします。

4. **[接続]** をクリックします。

#### 結果

管理ツールがデータベースに接続されます。

#### 次の手順

なし。

### 参照

- 「チュートリアル：サンプルデータベースへの接続」1 ページ
- 「[接続] ウィンドウを開く (Sybase Central の場合)」14 ページ

# 組み込みデータベースの接続

「組み込みデータベース」は、単一アプリケーションで使用するために設計されており、アプリケーションと同じコンピューター上で稼働し、通常はユーザーから隠されています。

アプリケーションが組み込みデータベースを使用する場合、アプリケーションの接続時には、通常、パーソナルサーバーは動作していません。データベースを起動するには、接続文字列を使用し、接続文字列の DatabaseFile (DBF) パラメーターでデータベースファイルを指定します。

自動的に起動されるデータベースのクエリのパフォーマンスを向上するには、ユーザーがすぐに接続しない場合でも、データベースをできるだけ早く起動します。このようにすることで、データベースに対してクエリが実行される前にキャッシュの準備が完了します。「キャッシュウォーミング」『SQL Anywhere サーバー SQL の使用法』を参照してください。

### ServerName (Server) 接続パラメーターの使用

組み込みデータベースを使用するときには、ServerName (Server) 接続パラメーターを使用します。これにより、同じコンピューター上で SQL Anywhere データベースサーバーを実行している別のアプリケーションが存在する場合でも、データベースが適切なデータベースサーバーに接続されます。

データベースサーバーが代替サーバー名を持っている場合、クライアントアプリケーションは、代替サーバー名を指定して起動したデータベースへの接続にのみ代替サーバー名を使用できます。そのデータベースサーバー上で実行中の他のデータベースに接続するのに代替サーバー名を使用することはできません。 「-sn dbsrv12 オプション」315 ページを参照してください。

### DatabaseFile (DBF) 接続パラメーターの使用

DBF 接続パラメーターは、使用するデータベースファイルを指定します。データベースファイルは、ServerName 接続パラメーターに一致するデータベースサーバーで (まだ起動していない場合は) 自動的に起動し、データベースサーバーは (まだ起動していない場合は) 自動的に起動されます。

データベースへの接続がなくなったら (通常は接続を開始したアプリケーションが切断したら)、データベースは自動的に停止します。データベースサーバーが自動的に起動された場合、データベースサーバーはデータベースが停止すると停止します。

次の例では、サンプルデータベースを組み込みデータベースとして起動しています。

```
"DBF=%SQLANYSAMP12%¥demo.db"
```

### StartLine [START] 接続パラメーターの使用

次の接続パラメーターは、組み込みデータベースとしてサンプルデータベースが開始されるのをカスタマイズする方法を示しています。StartLine 接続パラメーターは、データベースサーバーオプションを使用する場合に便利です。次に例を示します。

```
-c "START=dbeng12 -xd -c 8M;DBF=%SQLANYSAMP12%¥demo.db"
```

-xd オプションを指定すると、データベースサーバーがデフォルトのデータベースサーバーになりません。「-xd dbeng12/dbsrv12 サーバーオプション」292 ページを参照してください。

ユーザーが組み込みデータベースの起動や停止を行ったり、データベースサーバーの別のデータベースを起動したりしないようにするには、**none** に -gd オプションを設定します。「-gd dbeng12/dbsrv12 サーバーオプション」233 ページを参照してください。

接続パラメーターには、データベースサーバーの起動方法に影響を与えるものが多数存在します。StartLine (START) 接続パラメーターで対応するサーバーオプションを使用するのではなく、次の接続パラメーターを使用することをおすすめします。

- ServerName (Server)
- DatabaseFile (DBF)
- DatabaseSwitches (DBS)
- DatabaseName (DBN)

### Elevate 接続パラメーターの使用

Windows Vista でデータベースサーバーを自動的に起動する場合は、データベースサーバーの実行プログラムが昇格するよう、接続文字列で ELEVATE=YES を指定する必要があります。Windows Vista では、昇格されたデータベースサーバーのみが AWE メモリを使用したり、管理者ユーザーとしてプロシージャーを呼び出したりできます。

**注意**
AWE は使用されなくなりました。サイズの大きいキャッシュが必要な場合は、64 ビット版の Windows オペレーティングシステムで 64 ビット版の SQL Anywhere データベースサーバーを使用することをおすすめします。

### 例

次に示すのは、組み込みデータベースを使ったアプリケーションの完全な接続文字列の例です。

```
"UID=DBA;PWD=sql;DBF=c:¥myapp¥mydb.db;ServerName=myapp" "START=dbeng12 -c 12M -xd "
```

### 参照


- 「DatabaseFile (DBF) 接続パラメーター」333 ページ
- 「ServerName (Server) 接続パラメーター」365 ページ
- 「StartLine (START) 接続パラメーター」366 ページ
- 「-xd dbeng12/dbsrv12 サーバーオプション」292 ページ
- 「Elevate 接続パラメーター」340 ページ
- 「[接続] ウィンドウを開く (Sybase Central の場合)」14 ページ
- 「SQL Anywhere データベース接続」30 ページ


## SQL Anywhere ユーティリティの接続

SQL Anywhere のすべてのデータベースユーティリティで、データベースサーバーとの通信に Embedded SQL が使用されます。

### データベースユーティリティが接続パラメーターの値を取得する方法

多くの管理ユーティリティは、接続パラメーターの値を次の方法で取得します。

1. コマンドラインで指定された値を使用します。たとえば、次のコマンドは、ユーザー ID DBA とパスワード sql を使用して、デフォルトサーバー上のデフォルトデータベースのバックアップを開始します。

   ```
   dbbackup -c "UID=DBA;PWD=sql" c:¥backup
   ```

2. 値がない場合は、SQLCONNECT 環境変数の設定を使用します。SQL Anywhere では、この変数は自動的に設定されるわけではありません。

**参照**

- 「データベース管理ユーティリティ」909 ページ
- 「SQLCONNECT 環境変数」455 ページ

# SQL Anywhere データベース接続

データベースサーバー接続は、チャネルを形成します。クライアントアプリケーションからのアクティビティは、すべてそのチャネルを介して行われます。接続が確立されるまで、クライアントアプリケーションはデータベースサーバーと対話できません。データベースサーバーの接続が確立したら、ユーザーのパーミッションによって、ユーザーがデータベースサーバーに対して実行できる操作が決まります。

ユーザーがデータベースに接続すると、データベースサーバーはその接続にユニークな「接続 ID」 を割り当てます。データベースサーバーに対して新しく接続するたびに、サーバーは接続 ID 値を 1 つずつ増やします。これらの接続 ID は、データベースサーバーメッセージログに書き込まれます。このログには、情報メッセージ、エラー、警告、MESSAGE 文からのメッセージが記録されます。接続 ID を使用して、次のタスクを実行できます。

- 要求ログ情報のフィルタリング
- データベースにロックがある接続の識別
- データベースサーバー起動後の合計接続数やその接続の順番の追跡

CONNECTION_PROPERTY 関数を使用して Number 接続プロパティを問い合わせることによって、ユーザーの接続 ID を取得できます。

**参照**

- Number 接続プロパティ733 ページ
- 「データベースサーバー動作のロギング」153 ページ
- 「要求ロギング」『SQL Anywhere サーバー SQL の使用法』
- 「ロックの仕組み」『SQL Anywhere サーバー SQL の使用法』
- 「-z dbeng12/dbsrv12 サーバーオプション」296 ページ

# 接続パラメーターと接続文字列

クライアントアプリケーションはデータベースに接続するとき、一連の「接続パラメーター」を使用して接続を定義します。接続パラメーターは、データベースサーバー名、データベース名、

ユーザー ID などの情報を指定します。サポートされているすべての接続パラメーターのリストについては、「接続パラメーター」320 ページを参照してください。

各接続パラメーターでは、キーワードと値の組み合わせ (*parameter=value* の形式) を指定します。次の例では、デフォルトパスワードの password 接続パラメーターを指定しています。

```
PWD=sql
```

### 接続文字列として渡される接続パラメーター

接続パラメーターは、「接続文字列」にまとめられます。接続文字列では、次のように各接続パラメーターをセミコロンで区切ります。

```
parameter1=value1;parameter2=value2;...
```

次に例を示します。

```
UID=DBA;PWD=sql;Host=demo12;DatabaseName=demo;ServerName=myserver
```

通常、アプリケーションが構築してインターフェイスライブラリに渡す接続文字列は、ユーザーが情報を入力する方法と直接は対応していません。ユーザーが接続ウィンドウの複数のフィールドから入力することもあれば、アプリケーションが初期化ファイルから接続情報を読み込むこともあります。

SQL Anywhere のユーティリティの多くは、接続文字列を -c オプションとして認識し、変更を加えないでインターフェイスライブラリに渡します。次の例は、バックアップユーティリティ (dbbackup) の典型的なコマンドラインを示します。

```
dbbackup -c "HOST=myhost;DBN=demo;UID=DBA;PWD=sql;ServerName=demo12" SQLAnybackup
```

### マニュアルで接続文字列を表す

SQL Anywhere のマニュアルでは、接続文字列の例が、次の形式で表されることがあります。

```
parameter1=value1
parameter2=value2
...
```

この形式は、次の接続文字列と同じ意味を持ちます。

```
parameter1=value1;parameter2=value2
```

アプリケーションでは、接続文字列は、各パラメーター設定をセミコロンで区切って入力する**必要があります**。

### 参照


- 「一般的な接続シナリオ」32 ページ
- 「接続パラメーターの構文ルール」34 ページ
- 「接続パラメーター」320 ページ
- 「ネットワークプロトコルオプション」369 ページ
- 「文字セット変換」478 ページ

# 一般的な接続シナリオ

1 つのコンピューターでは、複数のデータベースサーバーを同時に実行できます。各データベースサーバーでは、複数のデータベースを同時に実行できます。データベース接続に必要な接続パラメーターは、複数のデータベースサーバーが同じコンピューターで実行されているか、複数のデータベースが同じデータベースサーバーで実行されているか、クライアントアプリケーションがデータベースサーバーとして同じコンピューターで実行されているかによって異なります。

次のリストは、いくつかの一般的な接続シナリオと、それぞれの場合に必要なクライアント接続パラメーターを示します。これらの接続パラメーターを組み合わせることで、接続先となるデータベースを正しく識別できます。次のリストの他にも、SQL Anywhere では、一般的ではない接続シナリオに対応するため、多くの接続パラメーターがサポートされています。

**注意**
コンピューターは、組み込みの SQL Anywhere サーバーを含むソフトウェアを実行している可能性があり、その場合、気づかずに複数のサーバーが実行されているかもしれません。常にサーバー名を指定する (ServerName=*server-name*) ことをおすすめします。

**ただ 1 つのデータベースサーバーでただ 1 つのデータベースが実行されており、クライアントアプリケーションと同じコンピューターで実行されている場合。** 必要な接続パラメーターは次のとおりです。

- 「Userid (UID) 接続パラメーター」
- 「Password (PWD) 接続パラメーター」
- 「ServerName (Server) 接続パラメーター」

```
dbisql -c "UID=DBA;PWD=sql;ServerName=Demo12"
```

**ただ 1 つのデータベースサーバーでただ 1 つのデータベースが実行されており、クライアントアプリケーションとは異なるコンピューターで実行されていて、データベースサーバーを実行しているコンピューターには myhost という名前のホストがある場合。**

- 「Userid (UID) 接続パラメーター」
- 「Password (PWD) 接続パラメーター」
- 「ServerName (Server) 接続パラメーター」
- 「Host 接続パラメーター」

```
dbisql -c "UID=DBA;PWD=sql;ServerName=demo12;Host=myhost"
```

**複数のデータベースサーバーでそれぞれ 1 つのデータベースが実行されており、クライアントアプリケーションと同じコンピューターで実行されている場合。**

- 「Userid (UID) 接続パラメーター」
- 「Password (PWD) 接続パラメーター」
- 「ServerName (Server) 接続パラメーター」

```
dbisql -c "UID=DBA;PWD=sql;ServerName=Demo12"
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**ただ 1 つのデータベースサーバーで複数のデータベースが実行されており、クライアントアプリケーションと同じコンピューターで実行されている場合。**

○「Userid (UID) 接続パラメーター」
○「Password (PWD) 接続パラメーター」
○「ServerName (Server) 接続パラメーター」
○「DatabaseName (DBN) 接続パラメーター」

```
dbisql -c "UID=DBA;PWD=sql;ServerName=demo12;DBN=demo"
```

**データベースサーバーまたはデータベースが実行中かどうかがわからず、それらを接続できるように、クライアントアプリケーションと同じコンピューター上で起動したいと思っている場合。**

○「Userid (UID) 接続パラメーター」
○「Password (PWD) 接続パラメーター」
○「ServerName (Server) 接続パラメーター」
○「DatabaseFile (DBF) 接続パラメーター」

```
dbisql -c "UID=DBA;PWD=sql;ServerName=Demo12;DBF=%SQLANYSAMP12%¥demo.db"
```

**参照**

● 「接続パラメーター」320 ページ

## デフォルト接続パラメーター

**注意**

●いくつかの接続パラメーターを指定しないで、デフォルトの動作で接続を確立できます。ただし、アプリケーションが他の SQL Anywhere アプリケーションとともにインストールされている場合は、運用環境でデフォルトの動作を使用すると問題が発生する場合があります。接続で問題がある場合、デフォルトに依存するのではなく、明示的に値を指定することで問題を解決できる場合があります。

●組み込みの SQL Anywhere サーバーを実行している場合は特に、常に ServerName 接続パラメーターを使用してサーバー名を指定することをおすすめします。「ServerName (Server) 接続パラメーター」365 ページを参照してください。

次のリストは、いくつかの一般的な接続シナリオと、それぞれの場合に必要なデフォルトクライアント接続パラメーターを示します。

### デフォルトを使用しない場合 (推奨方法)

次の接続文字列を使用して、名前を指定したデータベースで、名前を指定したローカルサーバーに接続します。

```
ServerName=server-name;DBN=db-name;UID=user-id;PWD=password
```

### デフォルトのデータベースサーバーとデータベース

ローカルコンピューターでただ 1 つのデータベースサーバーだけが実行中であることが確実なときに、接続にデフォルトパラメーターを使用します。

```
UID=user-id;PWD=password
```

### デフォルトのデータベースサーバー

ローカルコンピューターでただ 1 つのデータベースサーバーだけが実行中であることが確実なときには、デフォルトサーバー設定を使用して、接続するデータベースを指定できます。

```
DBN=db-name;UID=user-id;PWD=password
```

### デフォルトのデータベース

ローカルコンピューター上で複数のデータベースサーバーが実行中である場合、接続する先のサーバーを指定します。そのデータベースサーバーに 1 つのデータベースだけが実行中である場合は、データベース名を指定する必要はありません。次の接続文字列は、指定されたサーバーに接続して、デフォルトデータベースを使用します。

```
ServerName=server-name;UID=user-id;PWD=password
```

### ネットワークコンピューターのデフォルトのデータベースサーバー

次の接続文字列は、ネットワークコンピューター上で実行中のデフォルトデータベースサーバーで実行中のデータベースに接続します。この接続文字列は、データベースがデフォルトデータベース名を使用していることと、データベースサーバーがデフォルトの TCP/IP ポート 2638 上で実行していることを前提とします。

```
Host=host-name;DBN=db-name;UID=user-id;PWD=password;ServerName=server-name
```

データベースサーバーがデフォルトの TCP/IP ポート (2638) を使用していない場合、サーバーが Mac OS X 上の場合、またはデータベースサーバーが -sb オプションを使って起動された場合には、接続ができない可能性があります。これらの場合、Host=*host-name*:*port-number* を指定して接続できます。

## 接続パラメーターの構文ルール

- **スペースが含まれた接続文字列**　コマンドラインで接続文字列を指定するときに、接続パラメーター値にスペースが含まれている場合は接続文字列全体を二重引用符で囲む必要があります。
- **ブール値**　ブール (true または false) 引数は、true の場合は YES、ON、1、TRUE、Y、T のいずれかであり、false の場合は NO、OFF、0、FALSE、N、F のいずれかです。
- **大文字と小文字の区別**　値に大文字と小文字が区別されるもの (UNIX のファイル名など) が含まれている場合でも、接続パラメーター名は大文字と小文字を区別しません。
- **文字セット制限**　データベースサーバー名 (ServerName 接続パラメーターで指定) は、32 ～ 126 の範囲の ASCII 文字セットで構成してください。この制限は、他の接続パラメーターの値には適用されません。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

- **優先度**

> **警告**
> 接続文字列内では 1 つのパラメーターを複数回指定しないでください。パラメーターを複数回指定した場合、結果は保証されず、一貫しなくなる可能性があります。その上、動作はソフトウェアの将来のリリースで変更する可能性があります。

次のルールによって、接続パラメーターの優先度が決まります。

○接続文字列のエントリは、左から右に読み込まれます。同じ接続パラメーターが複数回指定された場合は、文字列の最後のパラメーターを適用します。ODBC と OLE DB のクライアントアプリケーション (Sybase Central、Interactive SQL、SQL Anywhere コンソールユーティリティは除く) はこの動作の例外です。これらに対して同じパラメーターが複数回指定された場合は、文字列の最初の値が適用されます。

○順番に、インターフェイスライブラリは、次のソースから接続パラメーターを取得します。

- **接続文字列**　パラメーターを接続文字列に明示的に渡すことができます。
- **SQLCONNECT 環境変数**　SQLCONNECT 環境変数に接続文字列を格納できます。
- **データソース**　ODBC データソースに接続パラメーターを格納できます。

  接続文字列と ODBC データソースがどちらも同じ接続パラメーターを指定する場合、接続文字列の値が使用され、データソースからの値は無視されます。

  たとえば、データベースサーバー demo12 がデータベース demo および demo2 を実行中であるとします。次のコマンドは demo データベースのデータソースを作成します。

  ```
  dbdsn -w demodsn -c "ServerName=demo12;Host=myhost;dbn=demo" -y
  ```

  接続文字列 DBN 値 (DBN=demo2) には、データソースの DBN 値よりも高い優先度があります。

  ```
  dbping -d -c "UID=DBA;PWD=sql;DBN=demo2;DSN=demodsn"
  ```

  接続に使用される接続パラメーターは次のとおりです。

  ```
  UID=DBA;PWD=sql;DBN=demo2;ServerName=demo12;Host=myhost
  ```

- **接続文字列の解析**　接続文字列の解析中に問題が発生した場合、問題の発生源が接続パラメーターであることを示すエラーが生成されます。
- **空の接続パラメーター**　指定されている値が空の接続パラメーターは、長さ 0 の文字列として扱われます。

**参照**


- 「接続パラメーター」320 ページ
- 「接続文字列と文字セット」478 ページ

# デスクトップアプリケーションと Windows Mobile データベース

Sybase Central や Interactive SQL など、デスクトップコンピューターで実行中のアプリケーションから、Windows Mobile デバイスで実行中のデータベースサーバーに接続できます。デスクトップコンピューターと Windows Mobile デバイスの間の Microsoft ActiveSync (Windows Mobile Device Center) リンクを介した接続では、TCP/IP を使用します。

**参照**


- 「Windows Mobile デバイス上でのデータベースサーバーの起動」411 ページ
- 「Windows Mobile デバイスに接続するための ODBC データソースの作成」412 ページ
- 「Windows Mobile デバイスの IP アドレスの特定」415 ページ


# ODBC データソース

Microsoft の「オープンデータベースコネクティビティ」(「ODBC」) は、クライアントアプリケーションを Windows ベースのデータベース管理システムに接続するための標準のアプリケーションプログラミングインターフェイスです。

多くのクライアントアプリケーションは、アプリケーション開発システムも含め、ODBC インターフェイスを使用して SQL Anywhere にアクセスします。データベースに接続するときは、ODBC アプリケーションは ODBC データソースを使用するのが一般的です。ODBC データソースは一連の接続パラメーターで、レジストリまたはファイルに格納されています。

**警告**
ユーザー ID、パスワード (暗号化の有無は不問)、データベースキーをデータソースに保存することはおすすめしません。「ODBC データソースにパスワードを含めない」1383 ページを参照してください。

SQL Anywhere ODBC ドライバーの名前は *dbodbc12.dll* で、*%SQLANY12%¥bin32* にあります。

次のアプリケーションから、ODBC データソースを使用して SQL Anywhere データベースに接続できます。

- Sybase Central、Interactive SQL、SQL Anywhere コンソールユーティリティ
- すべての SQL Anywhere ユーティリティ
- Sybase PowerDesigner Physical Data Model と InfoMaker
- Microsoft Visual Basic、Sybase PowerBuilder、Borland Delphi など、ODBC をサポートしているアプリケーション開発環境
- UNIX の SQL Anywhere クライアントアプリケーション。UNIX では、データソースはファイルとして保存されます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Sybase Open Client または jConnect とは ODBC データソースを使用できません。

**参照**

- 「ODBC 準拠」『SQL Anywhere サーバー プログラミング』

## ODBC データソースでの SQL Anywhere 接続パラメーター

ODBC データソースを使用してデータベースに接続することができます。ODBC データソースはクライアントコンピューター上にあります。

ODBC データソースには、一連の接続パラメーターが格納されています。SQL Anywhere の一連の接続パラメーターは、ODBC データソースとして、Windows レジストリに格納するか、ファイルとして格納できます。

SQL Anywhere では、ODBC インターフェイスを使用する Windows アプリケーション以外でも ODBC データソースを使用できます。

- UNIX と Windows の各オペレーティングシステムで動作する SQL Anywhere クライアントアプリケーションで ODBC データソースを使用できます。

- ODBC データソースは、jConnect と Sybase Open Client 以外のすべての SQL Anywhere クライアントインターフェイスで使用できます。UNIX と Windows Mobile の各オペレーティングシステムでは、データソースはファイルに格納されます。

データソースがある場合、使用するデータソースを次のように接続文字列で指定できます。

- **データソース名**　DataSourceName (DSN) 接続パラメーターを使用して、Windows レジストリ内のデータソースを参照します。

  DSN=my-data-source

- **ファイルデータソース**　FileDataSourceName (FILEDSN) 接続パラメーターを使用して、ファイルに格納されているデータソースを参照します。

  FileDSN=mysource.dsn

**注意**
接続文字列と ODBC データソースがどちらも同じ接続パラメーターを指定する場合、接続文字列の値が使用され、データソースからの値は無視されます。「接続パラメーターの構文ルール」34 ページを参照してください。

**参照**

- 「Windows でのファイルデータソース」45 ページ
- 「接続パラメーター」320 ページ
- 「ODBC データソースを使用した接続 (コマンドラインの場合)」18 ページ

# ODBC データソースの作成 (Sybase Central および Interactive SQL の場合)

ODBC データソースをすばやく作成するには、管理ツールの **[接続]** ウィンドウを使用します。

**前提条件**

データベースは、ローカルコンピューターで稼働しています。

**内容と備考**

次のとおりです。

**♦ ODBC データソースの作成 (Sybase Central および Interactive SQL の場合)**

1. **[接続]** ウィンドウを開きます。「[接続] ウィンドウを開く (Sybase Central の場合)」14 ページを参照してください。
2. 必要に応じて、**[データベースタイプを変更]** をクリックして、**[SQL Anywhere]** をクリックします。
3. **[ユーザー ID]** フィールドと **[パスワード]** フィールドは空白のままにします。

   **警告**
   ユーザー ID、パスワード (暗号化の有無は不問)、データベースキーをデータソースに保存することはおすすめしません。「ODBC データソースにパスワードを含めない」1383 ページを参照してください。

4. **[アクション]** ドロップダウンリストで、**[このコンピューターで稼働しているデータベースに接続]** をクリックし、コンピューター上で実行されているデータベースに接続します。
5. **[データベースファイル名]** を指定します。
6. **[ツール]** » **[ODBC データ・ソースとして保存]** をクリックします。
7. **[新しいデータソースの名前を入力]** フィールドにデータソース名を入力します。
8. **[データソースタイプを選択]** リストで、データソースを利用できるのは、現在のユーザーだけか、すべてのユーザーかを指定します。
9. (任意) 64 ビットバージョンの管理ツールを実行しており、32 ビットの ODBC ドライバーもインストールしている場合、**[32 ビットと 64 ビットの両方のデータソースを作成]** をクリックするオプションがあります。

   - システムデータソースを作成している場合、このオプションを選択して、32 ビットアプリケーションで使用できるデータソースと、64 ビットアプリケーションで使用できるデータソースの 2 つのシステムデータソースを作成します。
   - ユーザーデータソースを作成している場合、このオプションを選択して、一致するドライバーがインストールされている限りは 64 ビットと 32 ビットのアプリケーションの両方で

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

使用できる 1 つのデータソースを作成します。たとえば、このユーザーデータソースは、32 ビット ODBC ドライバーである *dbodbc12.dll* が %SQLANY12% にインストールされている限り、32 ビットアプリケーションで使用できます。また、64 ビット ODBC ドライバーである *dbodbc12.dll* が %SQLANY12% にインストールされている限り、64 ビットアプリケーションで使用できます。

10. **[保存]** をクリックします。

11. **[閉じる]** をクリックします。

**結果**

ODBC データソースが作成されます。

**次の手順**

なし。

**参照**

- 「ODBC データソースを使用した接続 (コマンドラインの場合)」18 ページ

## ODBC データソースの作成 (ODBC データソースアドミニストレーターの場合)

ODBC データソースアドミニストレーターを使用することで、ODBC データソースを作成することができます。

**前提条件**

なし。

### 内容と備考

**注意**

**64 ビット Windows でのシステム ODBC データソースの作成**　64 ビットバージョンの Windows では 64 ビットアプリケーション用と 32 ビットアプリケーション用の 2 つのシステムデータソースを管理します。32 ビットアプリケーションにアクセス可能なシステムデータソースを作成するには、32 ビット ODBC データソースアドミニストレーター (*WINDOWS¥System32* フォルダーにある) を実行する必要があります。接続の問題を避けるために、64 ビットのシステムデータソースと完全に同じになるように 32 ビットのシステムデータソースを設定してください。

**64 ビット Windows でのユーザー ODBC データソースの作成**　64 ビットバージョンの Windows は、1 セットのユーザーデータソースを管理します。ユーザーデータソースを作成するには、ODBC データソースアドミニストレーターの 32 ビットと 64 ビットのどちらのバージョンも使用できます。このデータソースは、一致するドライバーがインストールされている限り、64 ビットのアプリケーションでも 32 ビットのアプリケーションでも使用できます。たとえば、このユーザーデータソースは、32 ビット ODBC ドライバーである *dbodbc12.dll* が %SQLANY12% にインストールされている限り、32 ビットアプリケーションで使用できます。また、64 ビット ODBC ドライバーである *dbodbc12.dll* が %SQLANY12% にインストールされている限り、64 ビットアプリケーションで使用できます。

**♦ ODBC データソースを作成するには、次の手順に従います (ODBC データソースアドミニストレーターの場合)。**

1. **[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[ODBC データソースアドミニストレーター]** をクリックします。
2. 現在のユーザーの ODBC データソースを作成するには、**[ユーザー DSN]** タブをクリックします。

   システムデータソースを作成するには、**[システム DSN]** タブをクリックします。
3. **[追加]** をクリックします。
4. **[名前]** リストで **SQL Anywhere 12** を選択します。**[完了]** をクリックします。
5. ODBC データソースの接続パラメーターを指定します。

   **警告**
   ユーザー ID、パスワード (暗号化の有無は不問)、データベースキーをデータソースに保存することはおすすめしません。
6. **[OK]** をクリックします。
7. **[OK]** をクリックします。

### 結果

ODBC データソースが作成されます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 次の手順

なし。

### 参照

● 「ODBC データソースを使用した接続 (コマンドラインの場合)」18 ページ

## ODBC データソースの編集 (ODBC データソースアドミニストレーターの場合)

ODBC データソースを編集するには、ODBC データソースアドミニストレーターを使用する必要があります。

### 前提条件

ODBC データソースを定義する必要があります。

### 内容と備考

次のとおりです。

**♦ ODBC データソースを編集するには、次の手順に従います (ODBC データソースアドミニストレーターの場合)。**

1. **[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[ODBC データソースアドミニストレーター]** をクリックします。
2. **[ユーザー DSN]** または **[システム DSN]** のいずれかのタブを選択します。
3. **[名前]** リストでデータソースをクリックします。
4. **[設定]** をクリックします。
5. ODBC データソースの接続パラメーターを編集します。

   **警告**
   ユーザー ID、パスワード (暗号化の有無は不問)、データベースキーをデータソースに保存することはおすすめしません。

6. **[OK]** をクリックします。
7. **[OK]** をクリックします。

### 結果

ODBC データソースが編集され、保存されます。

### 次の手順

なし。

## ODBC データソースの作成 (dbdsn ユーティリティを使用する場合)

ODBC データソースの作成をスクリプト化するには、dbdsn ユーティリティを使用します。

### 前提条件

なし

### 内容と備考

dbdsn ユーティリティではファイルデータソースは作成できません。ファイルデータソースを作成するには、ODBC データソースアドミニストレーターを使用してください。システムデータソースは、Windows ベースのオペレーティングシステムに限られます。

**注意**

**64 ビット Windows でのシステムデータソースの作成**　64 ビットバージョンの Windows では 64 ビットアプリケーション用と 32 ビットアプリケーション用の 2 つのシステムデータソースを管理します。32 ビットアプリケーションにアクセスできるシステムデータソースを作成するには、32 ビット dbdsn ユーティリティ (SQL Anywhere インストールディレクトリの bin32 サブディレクトリにある) を実行する必要があります。接続の問題を避けるために、64 ビットのシステムデータソースと完全に同じになるように 32 ビットのシステムデータソースを設定してください。

**64 ビット Windows でのユーザーデータソースの作成**　64 ビットバージョンの Windows は、1 セットのユーザーデータソースを管理します。ユーザーデータソースを作成するには、dbdsn ユーティリティの 32 ビットと 64 ビットのどちらのバージョンも使用できます。このユーザーデータソースは、一致するドライバーがインストールされている限り、64 ビットのアプリケーションでも 32 ビットのアプリケーションでも使用できます。たとえば、このユーザーデータソースは、32 ビット ODBC ドライバーである *dbodbc12.dll* が %SQLANY12% にインストールされている限り、32 ビットアプリケーションで使用できます。また、64 ビット ODBC ドライバーである *dbodbc12.dll* が %SQLANY12% にインストールされている限り、64 ビットアプリケーションで使用できます。

**警告**
ユーザー ID、パスワード (暗号化の有無は不問)、データベースキーをデータソースに保存することはおすすめしません。

**♦ ODBC データソースを作成するには、次の手順に従います (dbdsn ユーティリティを使用する場合)。**

● データソースユーティリティ (dbdsn) を実行します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

たとえば、次のコマンドはサンプルデータベースの ODBC データソースを作成します。コマンドは、1 行に入力する必要があります。

```
dbdsn -w "My DSN" -c "DBF=$SQLANY12/demo.db"
```

### 結果

ODBC データソースが作成されます。

### 次の手順

なし。

### 参照

- 「データソースユーティリティ (dbdsn)」936 ページ
- 「ODBC データソースを使用した接続 (コマンドラインの場合)」18 ページ
- 「Windows でのファイルデータソース」45 ページ

## ODBC データソースの作成 (Mac OS X の場合)

Mac OS X で ODBC データソースを作成するには、ODBC データソースアドミニストレーターを使用します。

### 前提条件

ODBC データソースを作成する前に SQL Anywhere ODBC ドライバーをインストールします。「SQL Anywhere ODBC ドライバーの追加 (Mac OS x の場合)」44 ページを参照してください。

### 内容と備考

データソースユーティリティ (dbdsn) を使用して、Mac OS X で ODBC データソースを作成することも可能です。「データソースユーティリティ (dbdsn)」936 ページを参照してください。

**♦ ODBC データソースを作成するには、次の手順に従います (Mac OS X の場合)。**

1. */Applications/Utilities* から ODBC データソースアドミニストレーターを起動します。
2. ODBC データソースアドミニストレーターで、**[User DSN]** タブをクリックし、**[Add]** をクリックします。
3. **[名前]** リストで **[SQL Anywhere 12]** をクリックします。
4. **[OK]** をクリックします。
5. **[データソース名]** フィールドに、使用するデータソースのは前を入力します。
6. 次の接続パラメーターを追加します。接続パラメーターと値の大文字と小文字は区別されません。

| キーワード | 値 |
|---|---|
| UserID | DBA |
| Password | sql |
| StartLine | dbeng12 |
| ServerName | demo12 |
| ThreadManager | ON |
| Driver | SQL Anywhere 12 |

**警告**
ユーザー ID、パスワード (暗号化の有無は不問)、データベースキーをデータソースに保存することはおすすめしません。

接続パラメーターの詳細については、「接続パラメーター」320 ページを参照してください。

7. **[OK]** をクリックします。
8. **[Apply]** をクリックします。
9. [Command] キーを押しながら [Q] キーを押し、ODBC データソースアドミニストレーターを終了します。

**結果**

ODBC データソースが作成されます。

**次の手順**

なし。

**参照**


- 「ODBC データソースを使用した接続 (コマンドラインの場合)」18 ページ
- 「UNIX での ODBC データソース」48 ページ


## SQL Anywhere ODBC ドライバーの追加 (Mac OS x の場合)

ODBC データソースを作成する前に SQL Anywhere ODBC ドライバーをインストールします。「SQL Anywhere ODBC ドライバーの追加 (Mac OS x の場合)」44 ページを参照してください。

**前提条件**

必要であれば、http://mac.softpedia.com/progDownload/ODBC-Administrator-Tool-Download-61963.html にアクセスして ODBC データソースアドミニストレーターをダウンロードします。

### 内容と備考

次のとおりです。

#### ♦ SQL Anywhere ODBC ドライバーの追加 (Mac OS x の場合)

1. */Applications/Utilities* から ODBC データソースアドミニストレーターを起動します。
2. **[ドライバー]** タブをクリックします。
3. **[追加]** をクリックします。
4. **[Description]** フィールドに **SQL Anywhere 12** と入力します。
5. **[Choose]** をクリックし、**[Driver file]** と **[Setup file]** の両方のフィールドで SQL Anywhere ODBC ドライバーを選択します。デフォルトでは、*/Applications/SQLAnywhere12/System/lib32/dbodbc12_r.bundle* にあります。

   バンドル名の *_r* は、ドライバーのスレッドバージョンであることを示します。非スレッドアプリケーションで使用するための非スレッドバージョン (*dbodbc12.bundle*) もあります。
6. **[OK]** をクリックします。

   ODBC 設定ファイルは、ホームディレクトリの */Library/ODBC* にあります。ドライバー情報として *odbcinst.ini* ファイル、データソース情報として *odbc.ini* ファイルがあります。情報はテキストエディターを使用して編集できます。

### 結果

SQL Anywhere ODBC ドライバーが追加されます。

### 次の手順

なし。

### 参照

● 「ODBC データソースの作成 (dbdsn ユーティリティを使用する場合)」42 ページ

## Windows でのファイルデータソース

通常、Windows ベースのオペレーティングシステムでは、ODBC データソースをシステムレジストリに格納します。ファイルデータソースは、ファイルとして保存されるデータソースです。通常、Windows のファイルデータソースの拡張子は *.dsn* です。ファイルデータソースは複数のセクションから構成されていて、各セクションは角カッコで囲まれた名前で始まります。

ファイルデータソースを使用して接続するには、FileDataSourceName (FILEDSN) 接続パラメーターを使います。1 つの接続文字列内で DataSourceName (DSN) と FileDataSourceName (FILEDSN) の両方は使用できません。

正常に接続できない場合は、次のいずれかが発生します。

- **新しいファイルデータソースの場合**　Microsoft ODBC データソースアドミニストレーターは、指定のファイルデータソースパラメーターを使って接続を確立することができなかったというメッセージを表示します。その後、このメッセージから、検証されていないファイルを保存するかどうかが尋ねられます。ファイルデータソースを保存すると選択した場合は、ODBC データソースアドミニストレーターによって次の行のみが書き込まれます。

  ```
  [ODBC]
  DRIVER=SQL Anywhere 12
  ```

  ドライバー文字列は、SQL Anywhere のバージョンによって異なります。

- **既存のファイルデータソースの場合**　Microsoft ODBC データソースアドミニストレーターは、指定のファイルデータソース名が不正であることを示すメッセージを表示します。ファイルデータソースは、ODBC データソースアドミニストレーターによっては更新されません。

接続に成功すると、ODBC データソースアドミニストレーターによって新規または更新されたファイルデータソースがディスクに書き込まれますが、PWD=password パラメーターは含まれません (ファイルに以前に存在した場合は削除されます)。

**注意**
ファイルデータソースを使用すると、ファイルをユーザーに配布し、複数のユーザー接続の管理を簡単にすることができます。ファイルがファイルデータソースのデフォルト ロケーションにある場合、ODBC によって自動的に選択されます。

## ODBC ファイルデータソースの作成 (ODBC データソースアドミニストレーターの場合)

ODBC データソースアドミニストレーターを使用して、ODBC データソース情報をファイルに格納することができます。

### 前提条件

ファイルデータソースを正常に作成するには、ファイルデータソースの作成対象となるデータベースに対して接続を確立できる必要があります。

### 内容と備考

次のとおりです。

**♦ ODBC ファイルデータソースを作成するには、次の手順に従います (ODBC データソースアドミニストレーターの場合)。**

1. **[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[ODBC データソースアドミニストレーター]** をクリックします。

2. **[ファイル DSN]** タブをクリックします。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

3. **[追加]** をクリックします。
4. **[ドライバー]** リストで **SQL Anywhere 12** をクリックします。
5. **[次へ]** をクリックします。
6. **データ ソースの新規作成ウィザード**の指示に従います。
7. **[OK]** をクリックします。
8. **[OK]** をクリックします。

> **警告**
> ユーザー ID、パスワード (暗号化の有無は不問)、データベースキーをデータソースに保存することはおすすめしません。

**結果**

ODBC データソースファイルが作成されます。

**次の手順**

なし。

## Windows Mobile での ODBC データソース

Windows Mobile では、ODBC ドライバーマネージャーまたは ODBC データソースアドミニストレーターが提供されていません。Windows Mobile では、SQL Anywhere はファイルに格納されている ODBC データソースを使用します。これらのデータソース定義を使用するには、DSN または FILEDSN 接続パラメーターのどちらかを使用します。Windows Mobile では DSN と FILEDSN は同義語です。

> **警告**
> ユーザー ID、パスワード (暗号化の有無は不問)、データベースキーをデータソースに保存することはおすすめしません。

### データソースのロケーション

Windows Mobile は、デバイスのルートディレクトリ *¥filename.dsn* でデータソースファイルを検索します。

各データソースは 1 つのファイルに格納されます。このファイルはデータソースと同じ名前で、拡張子 *.dsn* が付きます。

### Windows Mobile データソースの例

次に、Windows Mobile 用の ODBC データソースの例を示します。

```
[ODBC]
DRIVER=¥windows¥dbodbc12.dll
```

```
Integrated=No
AutoStop=Yes
Host=192.168.0.55
LOG=¥sa_connection.txt
START=dbsrv12 -c 8M
SERVER=demo
```

**参照**


- 「Windows Mobile デバイスに接続するための ODBC データソースの作成」412 ページ
- 「Windows でのファイルデータソース」45 ページ


## UNIX での ODBC データソース

UNIX オペレーティングシステムでは、ODBC データソースはシステム情報ファイルに格納されます。このファイル名は通常 *.odbc.ini* です。SQL Anywhere データベースサーバーでは、次のロケーションでシステム情報ファイルが検索されます。

- ODBCINI 環境変数
- ODBC_INI 環境変数
- ODBCHOME 環境変数
- HOME 環境変数
- ユーザーのホームディレクトリ (~)
- PATH 環境変数

**注意**

ODBCINI と ODBC_INI 環境変数は、システム情報ファイル (ファイル名は通常 *.odbc.ini*) の検索に使用できますが、ODBCHOME と HOME 環境変数は *.odbc.ini* ファイルがあるパスの定義に使用できます。

ODBCINI と ODBC_INI はどちらも、ファイル名を含むフルパスで指定します。システム情報ファイルが ODBCINI または ODBC_INI で指定されたディレクトリにある場合、そのファイル名が *.odbc.ini* である必要はありません。

次にシステム情報ファイルの例を示します。

```
[My Data Source]
Host=hostname
ServerName=myserver
```

システム情報ファイルには、任意の接続パラメーターを入力できます。「接続パラメーター」320 ページを参照してください。

**警告**

ユーザー ID、パスワード (暗号化の有無は不問)、データベースキーをデータソースに保存することはおすすめしません。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

UNIX では、dbdsn ユーティリティを使用して ODBC データソースを作成、管理します。

**警告**
UNIX では、SQL Anywhere データソースだけを使用している場合を除き、ファイル難読化ユーティリティ (dbfhide) を使用して、システム情報ファイル (デフォルトのファイル名は *.odbc.ini*) を難読化しないでください。他のデータソース (Mobile Link 同期など) を使用する予定の場合、システム情報ファイルの内容を難読化すると、他のドライバーが正しく機能しなくなることがあります。

**参照**


- 「ODBC データソース」36 ページ
- 「データソースユーティリティ (dbdsn)」936 ページ
- 「ODBCHOME 環境変数 [UNIX]」447 ページ
- 「ODBCINI と ODBC_INI 環境変数 [UNIX]」447 ページ

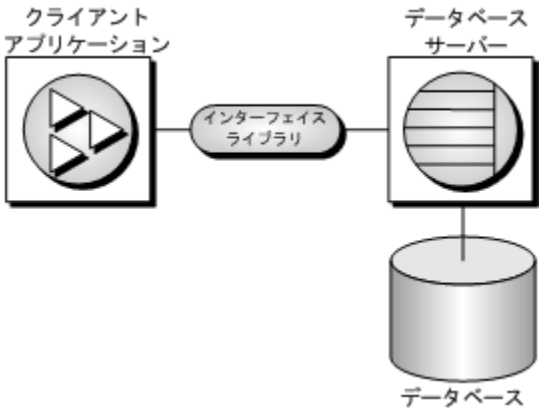

# SQL Anywhere API 接続

データベースに接続するには、クライアントアプリケーションで次のいずれかの SQL Anywhere API 関数を呼び出します。

| インターフェイス | 詳細 |
|---|---|
| ODBC | 「ODBC サポート」『SQL Anywhere サーバー プログラミング』<br><br>「ODBC データソース」 |
| OLE DB | 「OLE DB と ADO の開発」『SQL Anywhere サーバー プログラミング』 |
| ADO.NET | 「データベースへの接続」『SQL Anywhere サーバー プログラミング』 |
| Embedded SQL | 「Embedded SQL」『SQL Anywhere サーバー プログラミング』 |
| Sybase Open Client | 「Open Server としての SQL Anywhere」<br><br>「Sybase Open Client のサポート」『SQL Anywhere サーバー プログラミング』 |

| インターフェイス | 詳細 |
|---|---|
| SQL Anywhere JDBC ドライバー | 「JDBC クライアントアプリケーションからの接続」『SQL Anywhere サーバー プログラミング』<br><br>「JDBC サポート」『SQL Anywhere サーバー プログラミング』 |
| jConnect JDBC ドライバー | 「JDBC クライアントアプリケーションからの接続」『SQL Anywhere サーバー プログラミング』<br><br>「JDBC サポート」『SQL Anywhere サーバー プログラミング』 |

SQL Anywhere API では、クライアントアプリケーションからの呼び出しに含まれる接続情報を使用して、データベースサーバーを検出し、サーバーに接続します。クライアントアプリケーションから送信される情報には、ODBC データソース、SQLCONNECT 環境変数、またはデータベースサーバーアドレスキャッシュ (*sasrv.ini*) に格納されている情報があります。次の図に、このプロセスを簡単に示します。




## その他の情報

| 目的 | 参照先 |
|---|---|
| Sybase Central、Interactive SQL、SQL Anywhere コンソールユーティリティからの接続の概要 | 「[接続] ウィンドウを開く (Sybase Central の場合)」 |
| 高速起動の例 (Sybase Central と Interactive SQL のシナリオも含む) | 「SQL Anywhere データベース接続」 |
| データソースについて知りたい | 「ODBC データソース」 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 目的 | 参照先 |
|---|---|
| 使用できる接続パラメーターについて知りたい | 「接続パラメーター」 |
| 接続の確立方法について詳しく知りたい | 「接続のトラブルシューティング」 |
| ネットワーク固有の接続問題について知りたい | 「通信プロトコル」 |
| 接続に影響する文字セット問題について知りたい | 「接続文字列と文字セット」 |
| ファイアウォールを経由した接続について知りたい | 「ファイアウォール経由の接続」 |

**参照**

- 「高速の接続のための sasrv.ini でのデータベースサーバーアドレス情報のキャッシング」91 ページ
- 「SQLCONNECT 環境変数」455 ページ

# Windows 統合ログイン

Windows「統合化ログイン」機能を使用すると、オペレーティングシステム、ネットワークのログイン、データベース接続を、単一のユーザー ID とパスワードで管理できます。統合化ログインを作成するには、次の手順に従います。

- 統合化ログイン機能を有効にします。「統合化ログイン機能の有効化」52 ページを参照してください。
- 統合化ログインのマッピング先となるデータベースユーザーを作成します (存在しない場合)。
- Windows ユーザーまたはグループプロファイルと既存のデータベースユーザーの間に、統合化ログインマッピングを作成します。Sybase Central の **[ログインマッピング]** フォルダーに、統合化ログインパーミッションを持つユーザーがすべて表示されます。「ログインマッピング」53 ページを参照してください。
- クライアントアプリケーションから接続し、統合化ログイン機能をテストします。

## サポートされるオペレーティングシステム

統合化ログイン機能は、Windows ベースのデータベースサーバーで使用できます。Windows クライアントは、統合化ログインを使用して、Windows で実行されているネットワークサーバーに接続できます。

## 統合化ログインの利点

統合化ログインは、1 つまたは複数の Windows ユーザープロファイルまたは Windows ユーザーグループプロファイルから既存のデータベースユーザーへのマッピングです。ユーザープロ

ファイルまたはグループのセキュリティを特定し、コンピューターへのログインに成功したユーザーは、他のユーザー ID またはパスワードを指定しないでデータベースに接続できます。

そのためには、統合化ログインを使用するようにデータベースを設定し、コンピューターまたはネットワークへのログインに使用するユーザーまたはグループのプロファイルとデータベースユーザーの間のマッピングを許可します。

統合化ログインの使用は、ユーザーにとって便利であると同時に、1 つのセキュリティシステムでデータベースとネットワークの両方のセキュリティを維持できます。統合化ログインには次の利点があります。

- ユーザーによるユーザー ID とパスワードの入力は不要です。
- ユーザーはオペレーティングシステムによって認証されます。データベースのセキュリティと、コンピューターやネットワークのセキュリティには、単一のシステムが使用されます。
- 複数のユーザーまたはグループプロファイルを 1 つのデータベースユーザー ID にマッピングできます。
- Windows コンピューターへのログインに使用する名前とパスワードは、データベースユーザーの ID とパスワードと一致している必要はありません。

**警告**
統合化ログインにはセキュリティシステムが単一であるという利便性がありますが、そのためには、データベース管理者がセキュリティ上の重要事項を熟知しておく必要があります。「セキュリティについての考慮事項：無制限データベースアクセス」60 ページと「セキュリティについての考慮事項：コピーされたデータベースファイル」74 ページを参照してください。

## 統合化ログイン機能の有効化

login_mode データベースオプションは、統合化ログイン機能が有効かどうかを判断します。データベースオプションは、それが指定されているデータベースにしか適用されないため、同じサーバー内にロードされて動作している場合でも、データベースが異なれば統合化ログインの設定も異なります。

login_mode データベースオプションに指定できる値は次のとおりです。

- **Standard**　標準ログインを許可します。これがデフォルト設定です。標準ログインでは、ユーザー ID とパスワードの両方を入力する必要があり、Integrated や Kerberos の接続パラメーターは使用されません。統合化ログインまたは Kerberos ログインを使用して接続しようとすると、エラーが発生します。
- **Integrated**　統合化ログインを許可します。
- **Kerberos**　Kerberos ログインを許可します。
- **CloudAdmin**　このログインモードは、クラウドの内部で使用します。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**警告**
login_mode データベースオプションを標準ログインが許可されない設定にした場合、接続できるのは、統合化ログインマッピングまたは Kerberos ログインマッピングを付与されているユーザーまたはグループだけに制限されます。DBA 権限のあるユーザーでないかぎり、ユーザー ID とパスワードを使用して接続しようとすると、エラーが発生します。

複数のログインタイプを許可するには、login_mode オプションに複数の値を指定します。たとえば、次の SQL 文は、標準ログインと統合化ログインの両方の接続を許可するように login_mode データベースオプションの値を設定します。

```
SET OPTION PUBLIC.login_mode = 'Standard,Integrated';
```

データベースファイルがコピー可能な場合、一時的なパブリック login_mode オプションを使用する必要があります (統合化ログインと Kerberos ログインの両方で)。ファイルをコピーする場合、統合化ログインと Kerberos ログインはデフォルトではサポートされません。

**参照**

- 「Kerberos 認証」61 ページ

## ログインマッピング

SQL Anywhere では、ログインマッピングを作成して、Windows ユーザープロファイルまたは Kerberos プリンシパルをデータベース内の既存のユーザーにマッピングできます。どちらのタイプのログインマッピングも、**ログインマッピング作成ウィザード**または GRANT 文を使用して作成できます。ログインマッピングを作成するには DBA 権限が必要です。

**参照**

- 「統合化ログインのマッピング (Sybase Central の場合)」53 ページ
- 「Kerberos ログインマッピングの作成」67 ページ
- 「Windows 統合ログイン」51 ページ
- 「Kerberos 認証」61 ページ

## 統合化ログインのマッピング (Sybase Central の場合)

Windows 統合化ログインで SQL Anywhere を使用して、オペレーティングシステムとネットワークログインおよび SQL Anywhere データソース接続で、単一のユーザー ID とパスワードを維持することができます。

### 前提条件

統合化ログインマッピングを作成または削除するには、DBA 権限が必要です。

### 内容と備考

1 つのデータベースユーザー ID に対して 1 つのユーザーまたはグループプロファイルをマッピングする必要はありません。1 つ以上のユーザープロファイルを、同一のデータベースユーザー ID にマッピングできます。

ユーザープロファイルは、既存のデータベースユーザー ID に対してのみマッピングできます。データベースからデータベースユーザー ID が削除されると、そのデータベースユーザー ID に基づくすべての統合化ログインマッピングも、自動的に削除されます。

**♦ 統合化ログインをマッピングするには、次の手順に従います (Sybase Central の場合)。**

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。
2. 左ウィンドウ枠で **[ログインマッピング]** を右クリックし、**[新規]** » **[ログインマッピング]** をクリックします。
3. **[次へ]** をクリックします。
4. **[データベースに接続する Windows ユーザーを指定してください。]** フィールドに、統合化ログインを作成するユーザーまたはグループのプロファイル名を入力します。
5. **[Windows ユーザーに関連付けるデータベースユーザーを指定してください。]** リストで、このユーザーをマッピングするデータベースユーザー ID を選択します。
6. **ログインマッピング作成ウィザード**の残りの指示に従います。

### 結果

統合化ログインがマッピングされます。

### 次の手順

なし。

## 統合化ログインの作成 (SQL の場合)

Windows 統合化ログインで SQL Anywhere を使用して、オペレーティングシステムとネットワークログインおよび SQL Anywhere データソース接続で、単一のユーザー ID とパスワードを維持することができます。

### 前提条件

統合化ログインマッピングを作成または削除するには、DBA 権限が必要です。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 内容と備考

1 つのデータベースユーザー ID に対して 1 つのユーザーまたはグループプロファイルをマッピングする必要はありません。1 つ以上のユーザープロファイルを、同一のデータベースユーザー ID にマッピングできます。

**♦ 統合化ログインをマッピングするには、次の手順に従います (SQL の場合)。**

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. GRANT INTEGRATED LOGIN TO 文を実行します。

### 結果

統合化ログインが作成されます。

### 次の手順

なし。

### 例

次の SQL 文を使用すると、Window ユーザーの fran_whitney と matthew_cobb は、DBA ユーザー ID またはパスワードを知らなかったり指定しなかったりしても、DBA ユーザーとしてデータベースにログインできます。

```
GRANT INTEGRATED LOGIN
TO fran_whitney, matthew_cobb
AS USER DBA;
```

次の SQL 文を使用すると、Windows グループの mywindowsusers のメンバーである Windows ユーザーは、DBA ユーザーのユーザー ID またはパスワードを知らない、あるいは指定しなかった場合でも、DBA ユーザーとしてデータベースにログインできます。

```
GRANT INTEGRATED LOGIN
TO mywindowsusers
AS USER DBA;
```

### 参照

- 「Windows ユーザーグループの統合化ログイン」57 ページ
- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』

## 統合化ログインパーミッションの取り消し (Sybase Central の場合)

**♦ 統合化ログインパーミッションを取り消すには、次の手順に従います (Sybase Central の場合)。**

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。

2. 左ウィンドウ枠で **[ログインマッピング]** をクリックします。
3. 右ウィンドウ枠で、削除するログインマッピングを選択し、**[削除]** をクリックします。
4. **[はい]** をクリックします。

**♦ 統合化ログインパーミッションを取り消すには、次の手順に従います (SQL の場合)。**

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. REVOKE INTEGRATED LOGIN FROM 文を実行します。

**例**

次の SQL 文は、Windows ユーザーの pchin から統合化ログインパーミッションを削除します。

```
REVOKE INTEGRATED LOGIN
FROM pchin;
```

**参照**

- 「REVOKE 文」『SQL Anywhere サーバー SQL リファレンス』

## クライアントアプリケーションからデータベースへの接続

統合化ログインを使用してクライアントアプリケーションをデータベースに接続するには、次の処理を行います。

●接続パラメーターリストで Integrated (INT) パラメーターに YES を設定します。

●接続文字列または **[接続]** ウィンドウで、ユーザー ID もパスワードも指定しません。

接続文字列で Integrated (INT) パラメーターが YES に設定されている場合、統合化ログインが試行されます。接続が失敗し、login_mode データベースオプションが Standard,Integrated に設定されている場合、サーバーは標準ログインを試行します。「login_mode オプション」635 ページを参照してください。

ユーザー ID やパスワードを指定しないでデータベースに接続する場合、統合化ログインが試行されます。ログインの成否は、現在のユーザープロファイル名がデータベース内の統合化ログインマッピングに一致するかどうかによって決まります。

### Interactive SQL の例

次の例では、デフォルトのデータベースサーバーの統合化ログインマッピングと一致するユーザープロファイルでユーザーがログインした場合に接続に成功します。

```
CONNECT USING 'INTEGRATED=yes';
```

Interactive SQL の CONNECT 文は、次の場合にデータベースに接続できます。

●サーバーが現在実行中である。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

- デフォルトデータベースで、統合化ログイン接続を受け入れるよう login_mode データベースオプションが設定されている。
- 現在のユーザーのユーザープロファイル名と一致する統合化ログインマッピングまたはユーザーが所属する Windows ユーザーグループの統合化ログインマッピングが作成されている。
- 追加接続情報のプロンプトが表示されたときにユーザーが追加情報を入力しないで **[OK]** をクリックする。

## Windows ユーザーグループの統合化ログイン

Windows ユーザーがログインするとき、明示的な統合化ログインマッピングを持っていないが、統合化ログインマッピングがある Windows ユーザーグループに所属している場合、そのユーザーは、Windows ユーザーグループの統合化ログインマッピングで指定されたデータベースユーザーまたはグループとしてデータベースに接続します。

**警告**
Windows ユーザーグループの統合化ログインを作成すると、そのグループに属するすべてのユーザーが、ユーザー ID やパスワードを知らなくてもデータベースに接続できます。

「Windows ユーザーグループのメンバーによるデータベースへの接続を防ぐ」58 ページを参照してください。

### 複数グループのメンバー

Windows ユーザーが複数の Windows ユーザーグループに属し、コンピューター上の複数の Windows ユーザーグループの統合化ログインマッピングがデータベースに存在する場合、統合化ログインが成功するのは、コンピューター上の Windows ユーザーグループのすべてが同じデータベースユーザー ID に対する統合化ログインマッピングを持っている場合のみです。複数の Windows ユーザーグループが異なるデータベースユーザー ID への統合化ログインマッピングを持つ場合は、エラーが返され、統合化ログインは失敗します。

たとえば、dbuserA と dbuserB という 2 つのユーザー ID があるデータベースと、Windows ユーザーグループ xpgroupA と xpgroupB に属する Windows ユーザー windowsuser について考えてみます。

| SQL 文 | 許可内容 |
|---|---|
| GRANT INTEGRATED LOGIN<br>TO windowsuser<br>AS USER dbuserA; | windowsuser に対して明示的に設定された統合化ログインマッピングを使用して windowsuser はデータベースに接続します。 |
| GRANT INTEGRATED LOGIN<br>TO xpgroupA<br>AS USER dbuserB; | windowsuser は、xpgroupA に対して付与された統合化ログインマッピングを使用してデータベースに接続します。 |

| SQL 文 | 許可内容 |
|---|---|
| GRANT INTEGRATED LOGIN<br>TO xpgroupA<br>AS USER dbuserB;<br>GRANT INTEGRATED LOGIN xpgroupb<br>AS USER dbuserB; | windowsuser が属する両方の Windows ユーザーグループに同じデータベースユーザーへの統合化ログインマッピングがあるため、windowsuser はデータベースに接続できます。 |
| GRANT INTEGRATED LOGIN<br>TO xpgroupA<br>AS USER dbuserA;<br>GRANT INTEGRATED LOGIN<br>TO xpgroupb<br>AS USER dbuserB; | データベースへは接続できません。windowsuser がデータベースに接続しようとした場合、各 Windows ユーザーグループには異なるデータベースユーザーへの統合化ログインマッピングがあり、windowsuser は両方の Windows ユーザーグループのメンバーであるため、統合化ログインは失敗します。 |

### ドメインコントローラーのロケーション

デフォルトでは、Windows ユーザーグループのメンバーシップを確認するために、SQL Anywhere データベースサーバーを実行しているコンピューターが使用されます。ドメインコントローラサーバーが、データベースサーバーとは異なるコンピューターで実行されている場合は、integrated_server_name オプションを使用して、ドメインコントローラサーバーの名前を指定できます。次に例を示します。

```
SET OPTION PUBLIC.integrated_server_name = '¥¥myserver-1';
```

「integrated_server_name オプション」630 ページを参照してください。

## Windows ユーザーグループのメンバーによるデータベースへの接続を防ぐ

統合化ログインを持つ Windows ユーザーグループのメンバーであるユーザーが、グループの統合化ログインを使用してデータベースに接続できないようにするには、2 つの方法があります。

- パスワードのないデータベースユーザー ID に対して、ユーザーの統合化ログインを作成します。
- ユーザーがログインを許可されているかどうかを確認し、無許可のユーザーが接続しようとすると例外を発行するストアドプロシージャーを作成します。このストアドプロシージャーは、[login_procedure] オプションによって呼び出します。

### パスワードなしでユーザー ID への統合化ログインを作成する

ユーザーが統合化ログインを持つ Windows ユーザーグループのメンバーであるだけでなく、ユーザー ID に対する明示的な統合化ログインも持っている場合、データベースへの接続にはそ

のユーザーの統合化ログインが使用されます。ユーザーが Windows ユーザーグループ統合化ログインを使用してデータベースへ接続できないようにするには、データベースユーザー ID への Windows ユーザーの統合化ログインをパスワードなしで作成します。パスワードのないデータベースユーザー ID は、データベースに接続できません。

**♦ パスワードなしでユーザー ID への統合化ログインを作成するには、次の手順に従います (SQL の場合)。**

1. データベースにパスワードなしでユーザーを追加します。次に例を示します。

```
CREATE USER db_user_no_password;
```

2. パスワードを持たないデータベースユーザーにマッピングする Windows ユーザーの統合化ログインを作成します。次に例を示します。

```
GRANT INTEGRATED LOGIN TO WindowsUser
AS USER db_user_no_password;
```

### Windows ユーザーの接続を防ぐ手順の作成

login_procedure オプションは、データベースへの接続が試行されるたびに呼び出すストアドプロシージャーを指定します。デフォルトでは dbo.sp_login_environment プロシージャーが呼び出されます。login_procedure オプションを設定して、特定のユーザーがデータベースに接続できないようにするために作成したプロシージャーを呼び出すことができます。

次の例では、login_procedure オプションによって呼び出される login_check というプロシージャーを作成します。login_check プロシージャーは、データベースに接続できないユーザーのリストに照らし合わせて、指定されたユーザー名を確認します。指定されたユーザー名がリストに見つかると、接続は失敗します。この例では、Joe、Harry、または Martha というユーザーは接続を許可されていません。ユーザーがリストに見つからない場合、データベース接続は通常どおり実行され、sp_login_environment プロシージャーが呼び出されます。

```
CREATE PROCEDURE DBA.user_login_check()
  BEGIN
    DECLARE INVALID_LOGON EXCEPTION FOR SQLSTATE '28000';
    // Disallow certain users
    IF( CURRENT USER IN ('Joe','Harry','Martha') ) THEN
      SIGNAL INVALID_LOGON;
    ELSE
      CALL sp_login_environment;
    END IF;
  END
go
GRANT EXECUTE ON DBA.user_login_check TO PUBLIC
go
SET OPTION PUBLIC.login_procedure='DBA.user_login_check'
go
```

## ネットワークから見た統合化ログイン

データベースがネットワークサーバーにある場合、統合化ログインを使用するには次の 2 つの条件のどちらかが満たされていなければなりません。

- 統合化ログイン接続に使用するユーザープロファイルが、ローカルコンピューターとサーバーで同じである必要があります。両方のユーザープロファイルのパスワードも同じである必要があります。

  たとえば、ユーザー jsmith が統合化ログインを使用してネットワークサーバー上にロードされているデータベースに接続しようとした場合、ローカルコンピューターとデータベースサーバーを実行しているコンピューターとの両方に、同じユーザープロファイル名とパスワードが存在している必要があります。ユーザー jsmith は両方のコンピューターにログオンできる必要があります。

- ネットワークアクセスが Microsoft ドメインによって制御されている場合、統合化ログインを試みるユーザーは、ドメインコントローラーサーバーのドメインパーミッションを持っていて、ネットワークにログインしている必要があります。ローカルコンピューターのユーザープロファイルとネットワークサーバーのユーザープロファイルが一致する必要はありません。

## デフォルトの統合化ログインユーザー

デフォルトの統合化ログインユーザー ID を作成すると、現在使用されているユーザープロファイル用の統合化ログインマッピングが存在しなくても、統合化ログインを介した接続が成功します。

**例**

たとえば、ユーザープロファイル名 JSMITH に統合化ログインマッピングが存在しない場合、使用されているユーザープロファイルが JSMITH であれば、統合化ログイン接続をしようとしても通常は失敗します。

ただし、Guest という名前のユーザー ID をデータベースに作成すると、ユーザープロファイル JSMITH を明示的に識別する統合化ログインマッピングがない場合でも、統合化ログインは Guest ユーザー ID に正常にマッピングします。

**警告**
デフォルト統合化ログインユーザーは、データベースに Guest というユーザー ID が含まれている場合は、統合化ログインを試みるすべてのユーザーのデータベースへの接続を許可します。Guest ユーザー ID に対して付与されている権限によって、新たに接続したユーザーに対して付与されるパーミッションと権限が決定されます。

## セキュリティについての考慮事項：無制限データベースアクセス

統合化ログイン機能は、SQL Anywhere セキュリティシステムの代わりに Windows のログイン制御システムを使用しても動作し、ユーザー ID とパスワードを指定せずにデータベースに接続できます。データベースを実行しているコンピューターにログインできるユーザーは、原則的に、データベースセキュリティを通過します。

ユーザーが dsmith として Windows サーバーに正常にログインすると、統合化ログインマッピングかデフォルトの統合化ログインユーザー ID のどちらかがあれば、ID をさらに確認しなくても、データベースに接続できます。

統合化ログインを使用する場合、データベース管理者は、データベースへのアクセスを制限するために、Windows によって使用されるログインセキュリティに特に注意する必要があります。

**警告**
ユーザープロファイル Guest を有効にしておくと、そのサーバーが実行するデータベースには無制限アクセスが許可されます。

Guest ユーザープロファイルが有効で、かつパスワードがブランクの場合、そのサーバーに対するログインはすべて成功します。ユーザープロファイルがそのサーバーに存在することや、指定されたログイン ID にドメインログインパーミッションがあることは要求されません。事実上、何らかのログイン ID とパスワードを使用すれば、すべてのユーザーがサーバーにログインできます。デフォルトでは、Guest ユーザープロファイルにログインします。

統合化ログイン機能を有効にしてデータベースに接続する場合は、このことに注意する必要があります。

次の例では、データベースをホストしている Windows サーバーに、ブランクのパスワードで有効になる Guest ユーザープロファイルがあることが前提となっています。

- ユーザー fran_whitney とデータベースユーザー ID DBA との間に、統合化ログインマッピングが存在します。ユーザー fran_whitney が正しいログイン ID とパスワードを使用してサーバーに接続すると、完全な管理権限を持つユーザー DBA としてデータベースに接続されます。

  しかし、fran_whitney としてサーバーに接続しようとした他のユーザーも、指定したパスワードに関係なくサーバーにログインできます。これは、Windows がデフォルトで Guest ユーザープロファイルに接続しようとするためです。fran_whitney のログイン ID を使用してサーバーへのログインに成功すると、権限のないユーザーでも統合化ログインマッピングを使用して DBA としてデータベースに接続されます。

**注意**
セキュリティ確保のため Guest ユーザープロファイルを無効にします。統合化ログインの最も安全な方法は、SQL Anywhere データベースを実行するすべての Windows コンピューターで、Guest ユーザープロファイルを無効にすることです。これは、Windows のユーザーマネージャーユーティリティを使用して実行できます。

# Kerberos 認証

Kerberos ログイン機能を使用すると、データベース接続、オペレーティングシステム、ネットワークのログインを、単一のユーザー ID とパスワードで管理できます。Kerberos ログインの使用は、ユーザーにとって便利であると同時に、1 つのセキュリティシステムでデータベースとネットワークのセキュリティを維持できます。次のような利点があります。

- ユーザーはデータベースに接続するときにユーザー ID やパスワードを入力しなくてよい
- 複数のユーザーを 1 つのデータベースユーザー ID にマッピングできる
- Kerberos へのログインに使用する名前とパスワードはデータベースユーザーの ID とパスワードと一致している必要がない

Kerberos は、シークレットキー暗号方式を使用して強力な認証と暗号化を実現するネットワーク認証プロトコルです。Kerberos にログインしているユーザーは、ユーザー ID やパスワードを指定しなくてもデータベースに接続できます。

認証に Kerberos を使用できます。認証を Kerberos に委任するには、次の処理を行います。

- Kerberos ログインを使用するようにサーバーとデータベースを設定する。
- コンピューターまたはネットワークにログインするユーザー ID と、データベースユーザーの間にマッピングを作成する。

**警告**
Kerberos ログインを単一のセキュリティソリューションとして使用する場合に検討する必要があるセキュリティ上の重要事項があります。「セキュリティについての考慮事項：コピーされたデータベースファイル」74 ページを参照してください。

Kerberos ソフトウェアは SQL Anywhere に含まれません。このソフトウェアは別途入手してください。Kerberos ソフトウェアには次のコンポーネントが含まれます。

- **Kerberos ライブラリ**　Kerberos クライアントまたは GSS (Generic Security Services)-API ランタイムライブラリと呼ばれています。Kerberos ライブラリは、明確に定義されている GSS-API を実装したもので、Kerberos を使用するクライアントコンピューターおよびサーバーコンピューターでそれぞれ必要です。KDC として Active Directory を使用する場合は、サードパーティ製 Kerberos クライアントライブラリの代わりに、組み込みの Windows SSPI インターフェイスを使用できます。

  SSPI は、SQL Anywhere クライアントのみが Kerberos 接続パラメーターで使用できます。SQL Anywhere データベースサーバーは SSPI を使用できません。サポートされる SSPI 以外の Kerberos クライアントを使用する必要があります。

- **Kerberos キー配布センター (KDC) サーバー**　KDC はユーザーおよびサーバーの保管場所として機能します。また、ユーザーやサーバーの ID を検証します。KDC は、通常、アプリケーションやユーザーログインに使用しないサーバーコンピューターにインストールされます。

SQL Anywhere は、DBLib クライアント、ODBC クライアント、OLE DB クライアント、ADO.NET クライアント、Sybase Open Client、jConnect クライアントからの Kerberos 認証をサポートします。Kerberos 認証は SQL Anywhere トランスポートレイヤーセキュリティ暗号化と組み合わせて使用できますが、SQL Anywhere はネットワーク通信での Kerberos 暗号化をサポートしていません。

Windows では、Kerberos を Windows ドメインとドメインアカウントに使用します。Active Directory Windows Domain Controllers は Kerberos KDC を実装します。この環境で認証を行うには、データベースサーバーコンピューターにサードパーティ製 Kerberos クライアントまたはランタイムが必要ですが、Windows クライアントコンピューターはサードパーティ製 Kerberos クライアントまたはランタイムの代わりに組み込み Windows SSPI インターフェイスを使用できます。「Windows で Kerberos ログインに SSPI を使用する」68 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## Kerberos クライアント

Kerberos 認証がサーバープラットフォームで利用できます。テスト済みの Kerberos クライアントのリストについては、http://www.ianywhere.jp/tech/1061806-os_components.html を参照してください。

サポートされている Kerberos クライアントが使用する keytab ファイルと GSS-API ファイルのデフォルトの名前とロケーションのリストを次の表に示します。

**注意**
SSPI は、SQL Anywhere クライアントのみが Kerberos 接続パラメーターで使用できます。SQL Anywhere データベースサーバーは SSPI を使用できません。サポートされる SSPI 以外の Kerberos クライアントを使用する必要があります。

| Kerberos クライアント | デフォルト keytab ファイル | GSS-API ライブラリファイル名 | 説明 |
|---|---|---|---|
| Windows MIT Kerberos クライアント | *C:¥WINDOWS¥krb5kt* | *gssapi32.dll* または *gssapi64.dll* | KRB5_KTNAME 環境変数を設定してからデータベースサーバーを起動することで、別の keytab ファイルを指定できます。 |
| Windows CyberSafe Kerberos クライアント | *C:¥Program Files¥CyberSafe¥v5srvtab* | *gssapi32.dll* または *gssapi64.dll* | CSFC5KTNAME 環境変数を設定してからデータベースサーバーを起動することで、別の keytab ファイルを指定できます。 |
| UNIX MIT Kerberos クライアント | */etc/krb5.keytab* | *libgssapi_krb5.so* [1] | KRB5_KTNAME 環境変数を設定してからデータベースサーバーを起動することで、別の keytab ファイルを指定できます。 |
| UNIX CyberSafe Kerberos クライアント | */krb5/v5srvtab* | *libgss.so* [1] | CSFC5KTNAME 環境変数を設定してからデータベースサーバーを起動することで、別の keytab ファイルを指定できます。 |
| UNIX Heimdal Kerberos クライアント | */etc/krb5.keytab* | *libgssapi.so.1* [1] | |

[1] ファイル名はオペレーティングシステムと Kerberos クライアントバージョンによって異なります。

## Kerberos 認証の設定

**♦ SQL Anywhere データベースでの Kerberos 認証の設定**

1. クライアントとサーバーの両方に、Kerberos クライアントソフトウェアを GSS-API ランタイムライブラリも含めてインストールし、設定します。

   Active Directory KDC を使用している Windows クライアントコンピューターでは SSPI を使用できるので、Kerberos クライアントをインストールする必要はありません。「Windows で Kerberos ログインに SSPI を使用する」68 ページを参照してください。

2. 必要に応じて、Kerberos プリンシパルを各ユーザーの Kerberos キー配布センター (KDC) に作成します。

   Kerberos プリンシパルとは Kerberos ユーザー ID であり、形式は *user/instance@REALM* です。*/instance* はオプションです。Kerberos をすでに使用している場合は、プリンシパルはすでに存在しているので、各ユーザーに Kerberos プリンシパルを作成する必要はありません。

   プリンシパルの大文字と小文字は区別されるため、正しく指定する必要があります。大文字と小文字の違いしかない複数のプリンシパルのマッピングはサポートされていません (たとえば、jjordan@MYREALM.COM と JJordan@MYREALM.COM の両方にマッピングすることはできません)。

3. Kerberos プリンシパルを SQL Anywhere データベースサーバーの KDC に作成します。

   データベースサーバー用の Kerberos プリンシパルの形式は *server-name@REALM* です。*server-name* は SQL Anywhere データベースサーバー名です。プリンシパルでは大文字と小文字が区別されます。また、*server-name* ではマルチバイト文字や /、¥、@ は使用できません。以降の手順では、Kerberos プリンシパルが my_server_princ@MYREALM.COM であることを前提としています。

   サーバーでは KDC 認証に keytab ファイルが使用されるので、KDC 内にサーバーサービスプリンシパルを作成する必要があります。keytab ファイルは保護および暗号化されます。

4. セキュリティに十分注意しながら、プリンシパル *server-name@REALM* 用の keytab ファイルを KDC から抽出し、SQL Anywhere データベースサーバーを実行中のコンピューターにコピーします。keytab ファイルのデフォルトロケーションは、Kerberos クライアントとプラットフォームによって異なります。keytab ファイルのパーミッションは、SQL Anywhere サーバーが読み取ることができ、不正なユーザーに読み取りパーミッションがないよう設定する必要があります。

5. Kerberos を使用するように SQL Anywhere を設定します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## Kerberos を使用するように SQL Anywhere を設定

**♦ Kerberos を使用するよう SQL Anywhere データベースを設定するには、次の手順に従います。**

1. SQL Anywhere データベースで Kerberos 認証を設定します。「Kerberos 認証の設定」64 ページを参照してください。

2. SQL Anywhere サーバーを -krb または -kr オプションを使用して起動し、Kerberos 認証を有効にします。また、-kl オプションを使用して GSS-API ライブラリのロケーションを指定し、Kerberos を有効にすることもできます。

3. パブリックオプションまたは一時的なパブリックオプション login_mode を Kerberos を含む値に変更します。このオプションの設定を変更するには DBA 権限が必要です。Kerberos ログインを許可するかどうかは、login_mode データベースオプションで指定します。データベースオプションは、それが指定されているデータベースにしか適用されないため、同じサーバー内にロードされて動作している場合でも、データベースが異なれば Kerberos ログインの設定も異なります。次に例を示します。

   ```
   SET OPTION PUBLIC.login_mode = 'Kerberos,Standard';
   ```

   login_mode データベースオプションには、次の値を 1 つまたは複数指定できます。

   - **Standard**　標準ログインを許可します。これはデフォルト値です。標準ログインでは、ユーザー ID とパスワードの両方を入力する必要があり、Integrated や Kerberos の接続パラメーターは使用されません。
   - **Integrated**　統合化ログインを許可します。
   - **Kerberos**　Kerberos ログインを許可します。

   **警告**
   login_mode データベースオプションを Kerberos に設定すると、接続できるのは、Kerberos ログインマッピングを付与されているユーザーだけに制限されます。DBA 権限のあるユーザーでないかぎり、ユーザー ID とパスワードを使用して接続しようとすると、エラーが発生します。

4. クライアントが使用するデータベースユーザー ID を作成します。既存のデータベースユーザーに適切なパーミッションがあれば、そのデータベースユーザー ID を Kerberos ログインに使用できます。次に例を示します。

   ```
   CREATE USER "kerberos-user"
   IDENTIFIED BY abc123;
   ```

5. GRANT KERBEROS LOGIN TO 文を実行して、クライアントの Kerberos プリンシパルから既存のデータベースユーザー ID へのマッピングを作成します。この文の実行には DBA 権限が必要です。次に例を示します。

   ```
   GRANT KERBEROS LOGIN TO "pchin@MYREALM.COM"
   AS USER "kerberos-user";
   ```

マッピングのない Kerberos プリンシパルが使用されているときに接続する場合は、Guest データベースユーザー ID が存在し、パスワードが設定されていることを確認します。 「デフォルトの統合化ログインユーザー」60 ページを参照してください。

6. クライアントユーザーが Kerberos プリンシパルを使用してログオン済みである (有効な Kerberos TGT：Ticket Granting Ticket がある) こと、およびクライアントの Kerberos チケットの期限が切れていないことを確認します。ドメインアカウントにログインしている Windows ユーザーは、TGT をすでに持っており、プリンシパルに必要なパーミッションがあれば、サーバーに認証されます。

   チケット付与権限付きチケットはユーザーパスワードを使用して暗号化された Kerberos チケットで、ユーザー ID の検証に Ticket Granting Service が使用します。

7. KERBEROS 接続パラメーター (通常は KERBEROS=YES、ただし KERBEROS=SSPI または KERBEROS=*GSS-API-library-file* も使用可) を指定して、クライアントから接続します。ユーザー ID またはパスワードの接続パラメーターが指定された場合は、無視されます。次に例を示します。

```
dbisql -c "KERBEROS=YES;Server=my_server_princ"
```

### Interactive SQL の例

たとえば、次の Interactive SQL 文を使用した接続は成功します。ただし、接続が成功するためには、サーバーのデフォルトデータベース内の Kerberos ログインマッピングと一致するユーザープロファイル名を使用してユーザーがログインしていることが必要です。

```
CONNECT USING 'KERBEROS=YES';
```

次のすべてに該当する場合、Interactive SQL の CONNECT 文はデータベースに接続できます。

- サーバーが現在実行中である。
- 現在のサーバーのデフォルトデータベースが、Kerberos 認証接続を受け入れることができる。
- Kerberos ログインマッピングがユーザーの現在の Kerberos プリンシパルに対して作成されている。
- データベースサーバーから接続についての詳細情報を要求された場合 (Interactive SQL を使用した場合など) に、情報を追加せずに [OK] をクリックする。

### 参照


- 「-kl dbeng12/dbsrv12 サーバーオプション」250 ページ
- 「-kr dbeng12/dbsrv12 サーバーオプション (旧式)」252 ページ
- 「-krb dbeng12/dbsrv12 サーバーオプション」253 ページ
- 「login_mode オプション」635 ページ
- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「Kerberos ログインマッピングの作成」67 ページ


 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## Sybase Open Client または jConnect アプリケーションからの接続

Sybase Open Client または jConnect アプリケーションから接続するには、次の手順に従います。

●Kerberos 認証を設定します。「Kerberos 認証の設定」64 ページを参照してください。

●Kerberos を使用するように SQL Anywhere を設定します。「Kerberos を使用するように SQL Anywhere を設定」65 ページを参照してください。

●Adaptive Server Enterprise での Kerberos 認証と同様に、Sybase Open Client または jConnect を設定します。サーバー名は SQL Anywhere サーバー名である必要があります。名前の大文字と小文字は区別されます。Sybase Open Client または jConnect から代替サーバー名を使用して接続することはできません。

Kerberos プリンシパルの設定と keytab の抽出の詳細については、http://www.sybase.com/detail?id=1029260 を参照してください。

**参照**


- 「-krb dbeng12/dbsrv12 サーバーオプション」253 ページ
- 「-kr dbeng12/dbsrv12 サーバーオプション (旧式)」252 ページ
- 「-kl dbeng12/dbsrv12 サーバーオプション」250 ページ
- 「login_mode オプション」635 ページ
- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「CREATE USER 文」『SQL Anywhere サーバー SQL リファレンス』
- 「Kerberos (KRB) 接続パラメーター」350 ページ
- 「Kerberos 接続のトラブルシューティング」69 ページ


## Kerberos ログインマッピングの作成

**♦ Kerberos ログインマッピングを作成するには、次の手順に従います (Sybase Central の場合)。**

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。

2. 左ウィンドウ枠で **[ログインマッピング]** を右クリックし、**[新規]** » **[ログインマッピング]** をクリックします。

3. **ログインマッピング作成ウィザード**の指示に従います。

**♦ Kerberos ログインマッピングを作成するには、次の手順に従います (SQL の場合)。**

1. DBA 権限のあるユーザーとして、データベースに接続します。

2. GRANT KERBEROS LOGIN TO 文を実行します。

**参照**

- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』

**例**

次の SQL 文は、Windows ユーザーの pchin に KERBEROS ログインパーミッションを付与します。

```
GRANT KERBEROS LOGIN TO "pchin@MYREALM.COM"
AS USER "kerberos-user";
```

## Kerberos ログインパーミッションの取り消し

**♦ Kerberos ログインマッピングを取り消すには、次の手順に従います (Sybase Central の場合)。**

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。
2. 左ウィンドウ枠で **[ログインマッピング]** をクリックします。
3. 右ウィンドウ枠で、ログインマッピングを右クリックし、**[削除]** をクリックします。
4. **[はい]** をクリックします。

**♦ Kerberos ログインマッピングを取り消すには、次の手順に従います (SQL の場合)。**

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. REVOKE KERBEROS LOGIN FROM 文を実行します。

**参照**

- 「REVOKE 文」『SQL Anywhere サーバー SQL リファレンス』

**例**

次の SQL 文は、Windows ユーザーの pchin から KERBEROS ログインパーミッションを削除します。

```
REVOKE KERBEROS LOGIN
FROM "pchin@MYREALM.COM";
```

## Windows で Kerberos ログインに SSPI を使用する

Windows ドメインでは、クライアントコンピューターに Kerberos クライアントをインストールしなくても、Windows ベースのコンピューターで SSPI を使用できます。Windows ドメインアカウントには、関連付けられた Kerberos プリンシパルがあらかじめ用意されています。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### ♦ SSPI を使用した接続

**注意**
SSPI は、SQL Anywhere クライアントのみが Kerberos 接続パラメーターで使用できます。SQL Anywhere データベースサーバーは SSPI を使用できません。サポートされる SSPI 以外の Kerberos クライアントを使用する必要があります。

1. Kerberos 認証を設定します。 「Kerberos 認証の設定」64 ページを参照してください。

2. SQL Anywhere サーバーを -krb オプションを指定して起動し、Kerberos 認証を有効にします。次に例を示します。

   ```
   dbeng12 -krb -n my_server_princ C:¥kerberos.db
   ```

3. パブリックオプションまたは一時的なパブリックオプション login_mode を Kerberos を含む値に変更します。このオプションの設定には、DBA 権限が必要です。次に例を示します。

   ```
   SET OPTION PUBLIC.login_mode = 'Kerberos';
   ```

4. クライアントが使用するデータベースユーザー ID を作成します。(オプションで、既存のデータベースユーザーに適切なパーミッションがあれば、そのデータベースユーザー ID を Kerberos ログインに使用できます。) 次に例を示します。

   ```
   CREATE USER kerberos_user
   IDENTIFIED BY abc123;
   ```

5. クライアントの Kerberos プリンシパルから既存のデータベースユーザー ID へのマッピングを作成します。作成するには、GRANT KERBEROS LOGIN TO 文を実行します。この文の実行には、DBA 権限が必要です。次に例を示します。

   ```
   GRANT KERBEROS LOGIN TO "pchin@MYREALM.COM"
   AS USER "kerberos_user";
   ```

6. クライアントコンピューターからデータベースに接続します。次に例を示します。

   ```
   dbisql -c "KERBEROS=SSPI;Server=my_server_princ"
   ```

   接続文字列に Kerberos=SSPI と指定されている場合、Kerberos ログインが試行されます。

   次の Interactive SQL 文を使用した接続も成功します。ただし、接続が成功するためには、ユーザーは、サーバーのデフォルトデータベース内の Kerberos ログインマッピングと一致するユーザープロファイル名を使用してログオンしていることが必要です。

   ```
   CONNECT USING 'KERBEROS=SSPI';
   ```

## Kerberos 接続のトラブルシューティング

Kerberos 認証を有効にしようとしたり、使おうとしたりしたときに予期しないエラーが発生した場合は、データベースサーバーとクライアントで追加の診断メッセージを有効にすることをおすすめします。

データベースサーバーの起動時に -z オプションを指定します。または、すでに実行中のサーバーのデータベースサーバーメッセージログに追加の診断メッセージを表示するには CALL sa_server_option( 'DebuggingInformation', 'ON' ) を使用します。LogFile 接続パラメーターを使用すると、指定したファイルにクライアント診断メッセージが書き込まれます。 「-z dbeng12/dbsrv12 サーバーオプション」296 ページを参照してください。

LogFile 接続パラメーターを使用する代わりに、-z パラメーターを使用して Ping ユーティリティ (dbping) を実行することができます。-z パラメーターにより診断メッセージが表示され、接続の問題の原因を特定するのに役立ちます。「Ping ユーティリティ (dbping)」994 ページを参照してください。

## データベースサーバーの起動に関する問題

| 現象 | 一般的な解決策 |
| --- | --- |
| 「Kerberos GSS-API ライブラリをロードできません。」メッセージ | ●Kerberos クライアントが、GSS-API ライブラリも含めて、データベースサーバーコンピューターにインストールされていることを確認します。<br><br>●ロードしようとしたライブラリ名が、データベースサーバーの -z 出力にリストされます。ライブラリ名が正しいことを確認します。必要に応じて、-kl オプションを使用して正しいライブラリ名を指定します。<br><br>●フォルダーと、サポートしているライブラリが、ライブラリパス (Windows では %PATH%) にリストされていることを確認します。<br><br>●GSS-API ライブラリにエントリポイントがないことがデータベースサーバーの -z 出力に示されていた場合、そのライブラリはサポートされている Kerberos Version 5 GSS-API ライブラリではありません。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 現象 | 一般的な解決策 |
| --- | --- |
| 「サーバー名 "*server-name*" の Kerberos クレデンシャルを取得できません。」メッセージ | ●*server-name@REALM* のプリンシパルが KDC にあることを確認します。プリンシパルは大文字と小文字が区別されるので、データベースサーバー名とプリンシパル名のユーザー部分の大文字と小文字が一致していることを確認します。<br><br>●SQL Anywhere サーバー名がプリンシパルのプライマリ/ユーザー部分になっていることを確認します。<br><br>●サーバーのプリンシパルが keytab ファイルに抽出されていること、およびその keytab ファイルが Kerberos クライアントに対して適切なロケーションにあることを確認します。「Kerberos クライアント」63 ページを参照してください。<br><br>●データベースサーバーコンピューター上の Kerberos クライアントの領域がサーバープリンシパルの領域と異なっている場合は、-kr オプションを使用してサーバープリンシパルの領域を指定します。 |
| 「Kerberos ログインが失敗しました。」クライアントエラー | ●データベースサーバーの診断メッセージを確認します。サーバーが使用する keytab ファイルに関する問題のなかには、クライアントが認証しようとするまで検出されないものがあります。 |

### Kerberos クライアント接続のトラブルシューティング

クライアントが Kerberos 認証を使用して接続しようとしてエラーが発生した場合について、次の表に示します。

| 現象 | 一般的な解決策 |
| --- | --- |
| 「Kerberos ログインはサポートされていません。」エラーが発生し、LogFile にメッセージ 「Kerberos GSS-API ライブラリのロードに失敗しました」が出力されている。 | ●Kerberos クライアントが、GSS-API ライブラリも含めて、クライアントコンピューターにインストールされていることを確認します。<br><br>●LogFile で指定されたファイルに、ロードしようとしたライブラリがリストされています。ライブラリ名が正しいことを確認し、必要に応じて Kerberos 接続パラメーターを使用して正しいライブラリ名を指定します。<br><br>●サポートしているライブラリがあるディレクトリがライブラリパス (Windows では %PATH%) にリストされていることを確認します。<br><br>●GSS-API ライブラリにエントリポイントがないことが LogFile 出力に示されていた場合、そのライブラリはサポートされている Kerberos Version 5 GSS-API ライブラリではありません。 |
| 「Kerberos ログインはサポートされていません。」エラー | ●データベースサーバーに対して -krb、-kl、-kr の各サーバーオプションが 1 つ以上指定されており、Kerberos ログインが有効になっていることを確認します。<br><br>●Kerberos がクライアントとサーバーの両プラットフォーム上の SQL Anywhere でサポートされていることを確認します。「Kerberos クライアント」63 ページを参照してください。 |
| 「Kerberos ログインが失敗しました。」エラー | ●ユーザーが Kerberos にログイン済みであること、およびそのユーザーに期限が切れていない有効なチケット付与権限付きチケットがあることを確認します。<br><br>●クライアントコンピューターとサーバーコンピューターとの間で、時刻のずれが 5 分未満であることを確認します。 |
| 「ログインモード 'Kerberos' は、login_mode 設定で許可されていません。」エラー | ●Kerberos ログインが許可されるには、login_mode オプションのパブリックまたは一時的なパブリックのデータベースオプションの設定に値 Kerberos が含まれている必要があります。「login_mode オプション」635 ページを参照してください。 |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 現象 | 一般的な解決策 |
| --- | --- |
| "ログイン ID '*client-Kerberos-principal*' はどのデータベースユーザー ID にもマップされていません。" | ●Kerberos プリンシパルが GRANT KERBEROS LOGIN 文を使用してデータベースユーザー ID にマッピングされている必要があります。GRANT KERBEROS LOGIN 文には、領域を含む完全なクライアントプリンシパルが指定されている必要があります。また、インスタンスまたは領域しか違わないプリンシパルも別のプリンシパルとして扱われます。<br><br>●また、明示的にマッピングされていない有効な Kerberos プリンシパルを接続可能にするには、GRANT CONNECT を使用して、ゲストデータベースユーザー ID とパスワードを作成します。 |

## セキュリティについての考慮事項：一時的にパブリックオプションを設定してセキュリティを追加する

標準ログイン、統合化ログイン、Kerberos ログインの組み合わせを許可するよう、SET OPTION 文を使用してデータベースの login_mode オプションの値を設定すると、指定したログイン接続がそのデータベースで永続的に有効になります。たとえば、次の文は標準ログインと統合化ログインを永続的に許可します。

```
SET OPTION PUBLIC.login_mode = 'Standard,Integrated';
```

データベースを停止して再起動した場合でも、このオプションの値は変わらず、統合化ログインは有効のままです。

SET TEMPORARY OPTION を使用して login_mode オプションを設定した場合、統合化ログインによるユーザーアクセスは可能ですが、データベースがシャットダウンされるまでの間に限られます。次の文は、オプションの値を一時的に変更します。

```
SET TEMPORARY OPTION PUBLIC.login_mode = 'Standard,Integrated';
```

永久オプション値が Standard の場合、データベースは停止時にその値に戻ります。

一時的なパブリックオプションを設定すると、データベースのセキュリティを強化できます。統合化ログインまたは Kerberos ログインをデータベースに追加すると、データベースはそれを実行しているオペレーティングシステムのセキュリティに依存します。データベースを別のコンピューターにコピーすると、データベースへのアクセスは SQL Anywhere のセキュリティモデルに戻ります。

**参照**


- 「セキュリティについての考慮事項：コピーされたデータベースファイル」74 ページ
- 「SET OPTION 文」『SQL Anywhere サーバー SQL リファレンス』

## セキュリティについての考慮事項：コピーされたデータベースファイル

データベースファイルがコピー可能な場合、統合化ログインと Kerberos ログインに一時的なパブリック login_mode オプションを使用してください。ファイルをコピーした場合、統合化ログインと Kerberos ログインはデフォルトでサポートされません。

データベースに機密情報が含まれる場合、データベースファイルが保存されているコンピューターを不正アクセスから保護する必要があります。保護しなかった場合、データベースファイルがコピーされ、別のコンピューターからデータに不正にアクセスされる可能性があります。データベースのセキュリティを強化するには、次の処理を行います。

- パスワード、特に DBA 権限のあるユーザーパスワードを複雑にして、推測しにくくします。
- PUBLIC.login_mode データベースオプションを Standard に設定します。統合化ログインまたは Kerberos ログインを有効にする場合は、一時的なパブリックオプションのみがサーバーの起動ごとに変更されるようにします。これにより、データベースがコピーされたとしても、可能になるのは標準ログインだけになります。
- AES 暗号化アルゴリズムを使用して、データベースファイルを強力に暗号化します。暗号化キーは複雑にして、推測しにくくします。

**参照**

- 「セキュリティについての考慮事項：一時的にパブリックオプションを設定してセキュリティを追加する」73 ページ

# 接続プーリング

接続プールによって、データベースサーバーに対して複数の短時間接続を行うアプリケーションのパフォーマンスが向上する可能性があります。接続プールが接続に対して有効になっていると、接続は切断されたときに自動的にキャッシュされ、アプリケーションが再接続された時点で再利用できます。接続プールは、ConnectionPool (CPOOL) 接続パラメーターで制御します。同じ接続文字列を使用してアプリケーションで指定回数の接続を行うと、接続がプールされます。

接続をキャッシュするには、同じ接続文字列を使用してアプリケーションで最低でも 5 回の接続を行う必要があります。キャッシュされた接続を再利用するには、接続名はそのつどユニークでもかまいませんが、他のすべての接続パラメーターは同じにする必要があります。

アプリケーションで接続を再度処理するときに、同じ接続文字列に対して使用できる接続がキャッシュされている場合は、キャッシュされた接続が再利用されます。接続は、ConnectionPool (CPOOL) 接続パラメーターで指定された期間中、キャッシュされた状態のままになります (デフォルトでは 60 秒)。

アプリケーションの動作に変更があると、キャッシュされた接続は再利用されません。たとえば、キャッシュされた接続は、次の場合に再利用されません。

- 接続が存在しないときに自動的に停止するデータベースの場合。
- 接続が無効になっている場合。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

●データベースサーバーが接続制限値に達した場合。

●パスワードが変更された場合。

アプリケーションに対して接続プーリングを透過的にするため、接続のキャッシュ時に障害が発生した場合には接続が切断されます。キャッシュされた接続を再利用しようとするときに障害が発生すると、データベースサーバーで通常の接続が行われます。

接続が切断され、CPOOL 接続パラメーターで指定された最大接続数に達していない場合は、接続がキャッシュされます。接続が再初期化されます。接続はアプリケーションによって切断されますが、キャッシュされた接続はデータベースサーバーに接続したままになります。接続のクリーンアップと再初期化には、次の処理が含まれます。

●未処理のすべてのトランザクションのロールバック。

●テンポラリテーブル、テンポラリ関数、変数の削除。

●接続オプションと接続カウンターのリセット。

●データベースサーバーの接続カウントの増分と減分。キャッシュされた接続があるデータベースサーバーが停止した場合、アクティブな接続があることは通知されません。

●定義済みのすべての切断イベントと接続イベントの実行。

●login_procedure データベースオプションの実行とログインポリシーの検証。

●接続 ID のリセット。

### 他の接続プール製品での SQL Anywhere 接続プールの使用

接続プールをサポートしている製品または API を使用している場合、その製品または API の接続プールは SQL Anywhere の接続プールよりも優先されます。どちらのタイプの接続プールも、同時にアクティブにすることができます。

使用中の製品または API の接続プールの動作は、SQL Anywhere の接続プールと大幅に異なる場合があります。使用中の製品または API の接続プールの動作がアプリケーションに適していない場合は、SQL Anywhere の接続プールを使用して、アプリケーションのパフォーマンスを向上させることができます。

### 接続プールと読み込み専用のスケールアウト

接続に対して NodeType (NODE) 接続パラメーターも指定されている場合、通常、アプリケーションはプライマリサーバーに接続し、プライマリサーバーによって負荷が最も低いコピーノードが決まります。その時点で、そのノードに接続がリダイレクトされます。このような接続と切断をアプリケーションで短時間に何度も行うと、接続がプールされ、どのコピーサーバーを使用するかがプライマリサーバーに問われなくなります。この動作により、プライマリサーバーでの負荷は削減されますが、予期する動作にならない可能性があります。

**参照**

- 「ConnectionPool (CPOOL) 接続パラメーター」332 ページ
- 「接続パラメーターと接続文字列」30 ページ
- ConnPoolCachedCount データベースプロパティ787 ページ
- ConnPoolHits データベースプロパティ787 ページ
- ConnPoolMisses データベースプロパティ787 ページ

# テンポラリ接続

SQL Anywhere データベースサーバーでは、テンポラリ接続を使用して、バックアップの実行やデータベースの初期化などの操作が行われます。sa_conn_info または sa_conn_list システムプロシージャーを使用して、テンポラリ接続に関する情報を取得できます。ParentConnection プロパティでは、テンポラリ接続を作成した接続の接続 ID が返されます。ParentConnection 接続プロパティ735 ページを参照してください。

テンポラリ接続には、10 億 (1000000000) より大きい接続 ID と接続の機能を説明する名前があります。テンポラリ接続のリストについては、Name 接続プロパティ732 ページを参照してください。

次の例では、sa_conn_info システムプロシージャーを使用して、テンポラリ接続を作成した接続を示す結果セットが返されます。

```
SELECT Number, Name, ParentConnection FROM sa_conn_info();
```

接続 8 では、CREATE DATABASE 文を実行するテンポラリ接続が作成されます。

```
Number       Name              ParentConnection
-----------------------------------------------------------
1000000048   INT: CreateDB     8
9            SQL_DBC_14675af8  (NULL)
8            SQL_DBA_152d5ac0  (NULL)
```

**参照**

- 「sa_conn_info システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「sa_conn_list システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「CONNECTION_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』

# 接続のトラブルシューティング

SQL Anywhere で接続が確立される方法を理解すれば、接続の問題の解決に役立ちます。ファイアウォールを介した接続を含むネットワーク固有の問題については、「通信プロトコル」96 ページを参照してください。

SQL Anywhere では、接続を確立するときに次の処理が行われます。

1. インターフェイスライブラリを検出します。「インターフェイスライブラリの検出」77 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

2. 接続パラメーターリストをアセンブルします。「接続パラメーターリストのアセンブル」78 ページを参照してください。

3. データベースサーバーを検出します。データベースサーバーが検出されない場合は、SQL Anywhere はパーソナルサーバーを起動します。「データベースサーバーの検出」80 ページを参照してください。

4. データベースを検出します。「データベースの検出」91 ページを参照してください。

SQL Anywhere の接続手順は、次の場合も同じです。

- SQLDriverConnect 関数を使用する「ODBC アプリケーション」。これは ODBC アプリケーションでの一般的な接続方法です。Sybase PowerBuilder などのアプリケーション開発システムの多くは、このクラスのアプリケーションに属します。ODBC アプリケーションでは SQLConnect 関数も使用できます。

- 「Embedded SQL」を使用し、データベースとの接続で推奨関数 (db_string_connect) を使用するクライアントアプリケーション。さらに、Embedded SQL アプリケーションと Interactive SQL では、SQL CONNECT 文も使用できます。これには、CONNECT AS と CONNECT USING の 2 つの書式があります。dbisqlc を含むすべてのデータベースユーティリティは、db_string_connect を使用します。

- ADO.NET を使用する任意の 「.NET」 アプリケーション。アプリケーションは新しい SAConnection オブジェクトを作成し、接続文字列をコンストラクターに渡すか、ConnectionString プロパティを設定します。次に、アプリケーションは SAConnection オブジェクトの Open メソッドを呼び出して接続します。

- ADODB Connection オブジェクトを使用する任意の「ADO アプリケーション」。Provider プロパティは、OLE DB ドライバーの場所を指定するために使用されます。ConnectionString プロパティでは、**DataSourceName** の代わりに **DataSource** が使用され、**UserID** の代わりに **User ID** が使用されることがあります。

- 「SQL Anywhere JDBC ドライバー」を使用して、Driver Manager.GetConnection メソッドのパラメーターとして URL **jdbc:sqlanywhere:** の後に標準の接続文字列を渡すアプリケーション。

### 参照


- 「データベースサーバー起動時のトラブルシューティング」1211 ページ
- 「ネットワーク通信のトラブルシューティング」1215 ページ


## インターフェイスライブラリの検出

通常、この DLL または共有ライブラリのロケーションは、ユーザーには見えません。

### ODBC ドライバーのロケーション

ODBC では、インターフェイスライブラリは ODBC ドライバーとも呼ばれます。ODBC クライアントアプリケーションが ODBC ドライバーマネージャーを呼び出し、ドライバーマネージャーが SQL Anywhere ドライバーを検出します。

ODBC ドライバーマネージャーは、ドライバーを探すために指定されたデータソースを調べます。ODBC データソースアドミニストレーターまたは dbdsn ユーティリティを使用してデータソースを作成すると、SQL Anywhere は ODBC ドライバーに現在のロケーションを入力します。データソース情報は、Windows の場合はレジストリに格納され、UNIX の場合はシステム情報ファイル (デフォルトのファイル名は *.odbc.ini*) に格納されます。

### Embedded SQL インターフェイスライブラリのロケーション

Embedded SQL アプリケーションは、通常は SQL Anywhere に組み込まれた SQL インポートライブラリに対してリンクされます。インターフェイスライブラリは、アプリケーションの起動時に名前でロードされます。SQL Anywhere の Embedded SQL インターフェイスライブラリの名前は次のとおりです。

- **Windows** *dblib12.dll*
- **UNIX** *libdblib12* (オペレーティングシステム固有の拡張子が付きます)

### OLE DB ドライバーのロケーション

SQL Anywhere の OLE DB プロバイダー DLL (*dboledb12.dll*) の検索には、レジストリのエントリに基づいた、プロバイダー名 (SAOLEDB) が使用されます。エントリは、SAOLEDB のインストール時、または再登録時に作成されます。

### ADO.NET

ADO.NET プログラムは SQL Anywhere ADO.NET プロバイダー (*iAnywhere.Data.SQLAnywhere.dll*) への参照を追加します。.NET データプロバイダー DLL は、インストール時に .NET グローバルアセンブリキャッシュ (GAC) に追加されます。

### SQL Anywhere JDBC ドライバーのロケーション

アプリケーションの実行時は、Java パッケージ *sajdbc4.jar* (JDBC 4.0) がクラスパスに含まれていることが必要です。システムでネイティブ DLL または共有オブジェクトを検出できる必要があります。

- **Windows** Windows では、現在のディレクトリ、システムパス、*Windows* ディレクトリ、*Windows¥system32* ディレクトリが検索されます。
- **UNIX** UNIX では、システムパスとユーザーライブラリパスが検索されます。

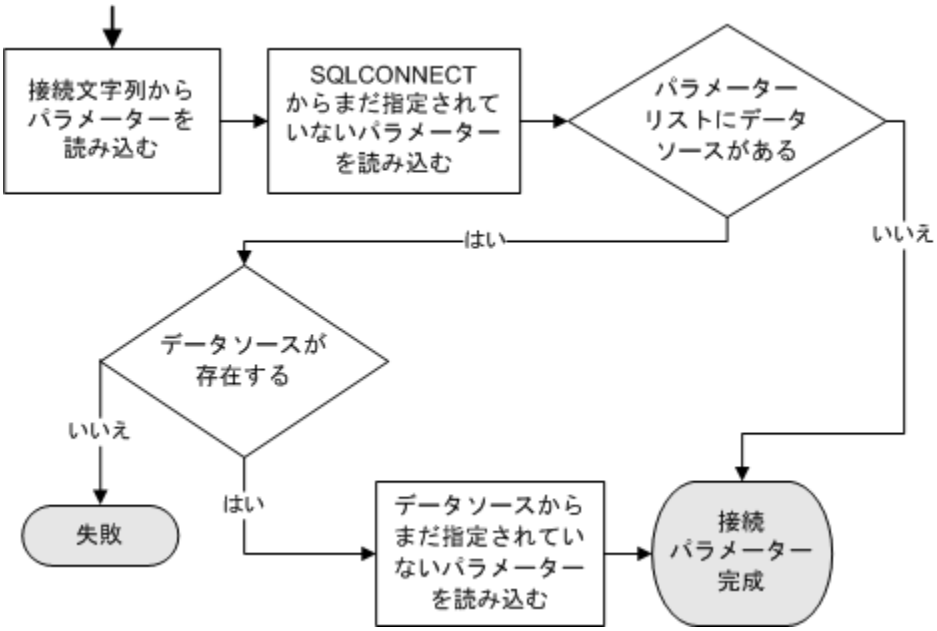
### ライブラリが検出されるタイミング

クライアントアプリケーションでインターフェイスライブラリが検出されたらインターフェイスライブラリに接続文字列が送信されます。この文字列は、接続パラメーターのリストをアセンブルし、サーバー接続を確立するためにインターフェイスライブラリで使用されます。

## 接続パラメーターリストのアセンブル

次の図は、インターフェイスライブラリが接続パラメーターのリストをアセンブルし、接続を確立する方法を示しています。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1




● **優先度**　複数の場所に格納されているパラメーターは次の優先順位に従います。

1. 接続文字列
2. SQLCONNECT 環境変数
3. ODBC データソース

> **注意**
> 接続文字列と ODBC データソースがどちらも同じ接続パラメーターを指定する場合、接続文字列の値が使用され、データソースからの値は無視されます。「接続パラメーターの構文ルール」34 ページを参照してください。

● **失敗**　この段階での失敗は、接続文字列が不正な構文を使用している場合、不明な接続パラメーターを使用している場合、または、接続文字列または SQLCONNECT で存在しないデータソースを指定した場合に発生する可能性があります。

● **無視されるパラメーター**　すでに使用中である他の接続に従い、次のようないくつかの接続パラメーターは無視される可能性があります。

○ **Autostop**　データベースがすでにロードされている場合は無視されます。

○ **DatabaseFile**　DatabaseName が指定され、この名前を持つデータベースがすでに実行されている場合は無視されます。

インターフェイスライブラリは、アセンブル後の接続パラメーターのリストを使用して接続を試行します。

**参照**

- 「接続パラメーターの構文ルール」34 ページ

## データベースサーバーの検出

SQL Anywhere は実行中のサーバーを検出したときに、必要なデータベースをそのサーバー上で検出または起動しようとします。「データベースの検出」91 ページを参照してください。

SQL Anywhere は、実行中のサーバーを検出できないと、接続パラメーターに応じてパーソナルサーバーを起動しようとする場合があります。

**注意**

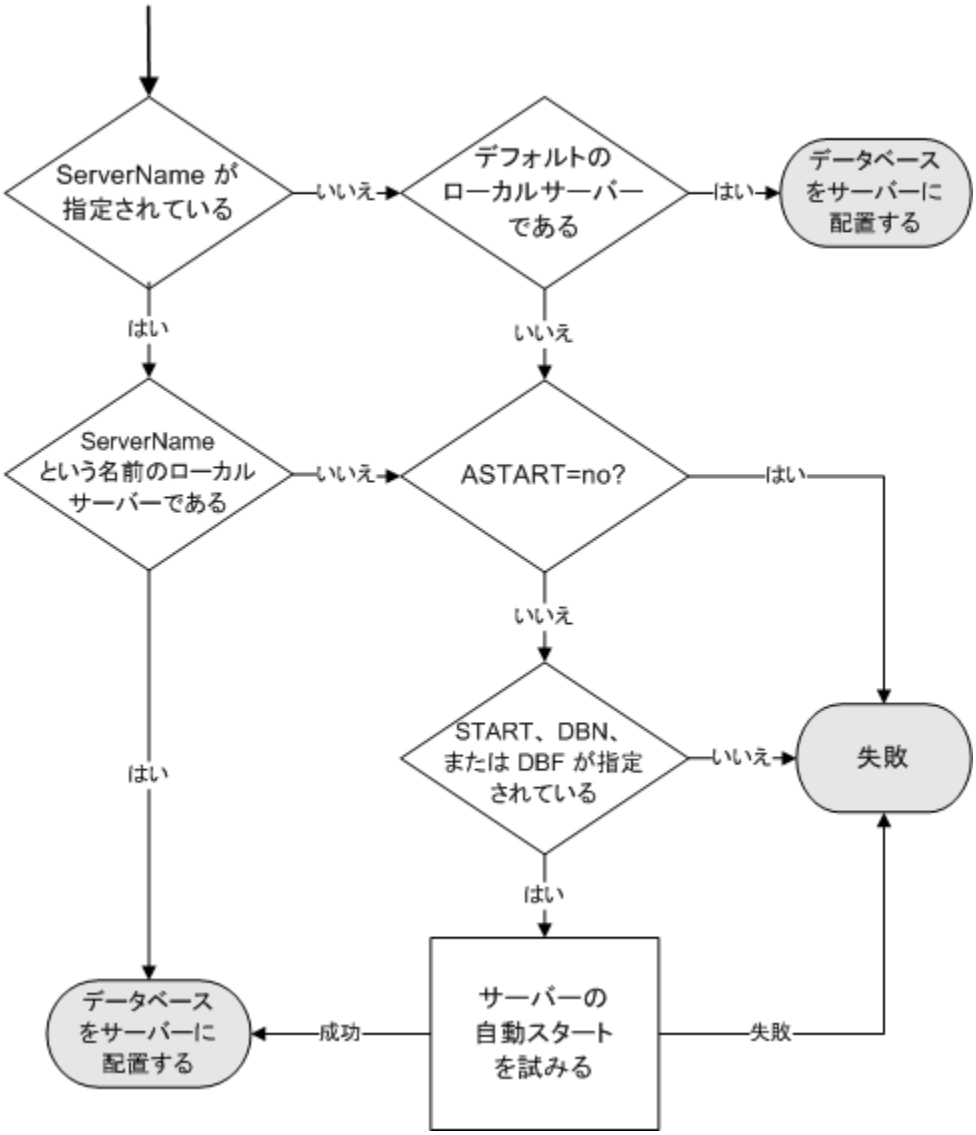
- ローカル接続では、サーバーの検出は簡単です。ネットワーク経由の接続では、Host 接続パラメーターを使用することをおすすめします。
- クライアントがネットワークサーバーを検出する方法の詳細については、「Host 接続パラメーターを使用した SQL Anywhere データベースサーバーの検出」84 ページ を参照してください。
- サーバーが自動的に起動される場合は、START、DBF、DBKEY、DBS、DBN、Server、および AutoStop 接続パラメーターの情報を使用して、そのサーバーのオプションが構築されます。サーバーは、Host または CommLinks=TCPIP 接続パラメーターが指定されている場合は、自動的には起動しません。
- サーバーに代替サーバー名がある場合、代替サーバー名を指定して起動したデータベースへの接続にのみ代替サーバー名を使用できます。そのデータベースサーバー上で実行中の他のデータベースに接続するのに代替サーバー名を使用することはできません。AS PRIMARY 句、CREATE MIRROR SERVER 文『SQL Anywhere サーバー SQL リファレンス』と「-sn dbsrv12 オプション」315 ページを参照してください。

次の図は、SQL Anywhere がデータベースサーバーを検出する方法を示しています。

### 接続文字列が CommLinks 接続パラメーターまたは Host 接続パラメーターを含まない

次の図は、接続文字列が CommLinks 接続パラメーターまたは Host 接続パラメーターを含まないときに、SQL Anywhere が共有メモリを使ってデータベースサーバーを検出する方法を示しています。

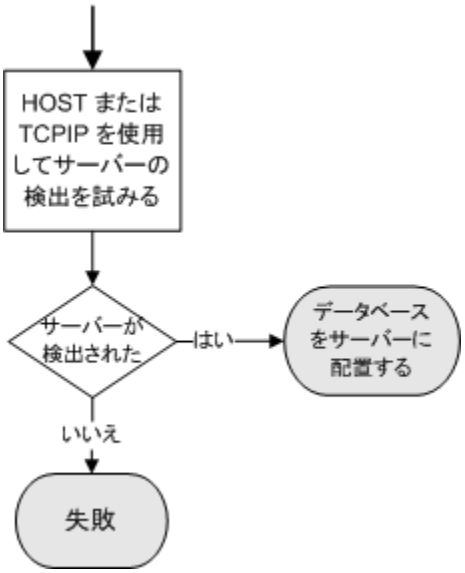
 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1




SQL Anywhere は、実行中のサーバーを検出できないと、接続パラメーターに応じてパーソナルサーバーを起動しようとする場合があります。

### Host 接続パラメーターまたは CommLinks 接続パラメーターを含む接続文字列は TCPIP を指定する

次の図は、接続文字列に CommLinks 接続パラメーターまたは Host 接続パラメーターのどちらかが含まれ、TCPIP を指定するときに、SQL Anywhere がデータベースサーバーを検出する方法を示しています。




**接続文字列には、ShMem および TCPIP に設定されている CommLinks 接続パラメーターが含まれている**

接続文字列に CommLinks=ShMem、TCPIP または CommLinks=TCPIP、ShMem が含まれている場合、SQL Anywhere は次の処理を行います。

- データベースサーバーの自動起動を除く、共有メモリ接続を実行します。「接続文字列が CommLinks 接続パラメーターまたは Host 接続パラメーターを含まない」80 ページを参照してください。

  SQL Anywhere は、共有メモリを使用したサーバーの検出に失敗した場合、TCPIP 接続を行おうとします。「Host 接続パラメーターまたは CommLinks 接続パラメーターを含む接続文字列は TCPIP を指定する」81 ページを参照してください。

**参照**


- 「-xd dbeng12/dbsrv12 サーバーオプション」292 ページ
- 「サーバー列挙ユーティリティ (dblocate)」999 ページ
- 「db_string_ping_server 関数」『SQL Anywhere サーバー プログラミング』
- 「Host 接続パラメーターを使用した SQL Anywhere データベースサーバーの検出」84 ページ
- 「CommLinks=TCPIP 接続パラメーターを使用した SQL Anywhere データベースーサーバーの検出」87 ページ


## Broadcast Repeater ユーティリティを使用したデータベースサーバーの検出

Broadcast Repeater ユーティリティ (dbns12) を使用すると、他のサブネット上で実行されている SQL Anywhere データベースサーバーや、ファイアウォールの外側にあって UDP ブロードキャス

トが通常は届かない SQL Anywhere データベースサーバーを、SQL Anywhere クライアントは Host 接続パラメーターや LDAP を使用することなく検索できます。

### 前提条件

この作業を実行するための前提条件はありません。

### 内容と備考

相互通信が可能な DBNS プロセスの数に制限はありません。各 DBNS プロセスが検出したすべての DBNS プロセスに接続し、複数の DBNS プロセスが DBNS プロセスリストを共有します。たとえば、A と B の 2 つの DBNS プロセスを起動するとします。第 3 のサブネットで 3 番目の DBNS プロセス C を起動し、プロセス B のアドレスをプロセス C に渡すと、プロセス B がプロセス C にプロセス A の存在を通知し、プロセス C がプロセス A に接続します。

**注意**
1 つのサブネットで複数の DBNS プロセスを実行することは一般には不要であり、推奨できません。

1 つのサブネット内で複数の DBNS プロセスを実行しなければならない場合は、各 DBNS プロセスに対して異なるポートを指定する必要があります。1 つの DBNS プロセスが受信できるのは 1 つのポートからだけです。Broadcast Repeater ユーティリティ (dbns12) で -ap パラメーターを使用します。

Host 接続パラメーターまたは HOST プロトコルオプションのいずれかが使用される場合は、Broadcast Repeater ユーティリティは必要ありません。

### ♦ タスク

1. サブネット内の任意のコンピューターで DBNS (データベースネームサービス) プロセスを起動します。

2. 別のサブネット内の任意のコンピューターで DBNS プロセスを起動し、最初のコンピューターのコンピューター名または IP アドレスをパラメーターとして渡します。

   2 つの DBNS プロセスが TCP/IP 相互接続を確立します。

3. いずれの DBNS プロセスもそれぞれのサブネット上でブロードキャストを受信します。各 DBNS プロセスが TCP/IP 接続上で他方の DBNS プロセスに要求を送信し、受信した DBNS プロセスはそのサブネット上で要求を再ブロードキャストし、さらに送信元の DBNS プロセスに応答を送信します。送信元の DBNS プロセスは、受信した応答を要求元のクライアントに送信します。

### 結果

いずれかのサブネット上の通常の SQL Anywhere ブロードキャストは、リモートサブネット上のデータベースサーバーに到達し、クライアントは HOST パラメーターを指定しなくてもリモートサブネット上のデータベースサーバーに接続できます。

**次の手順**

なし。

**参照**

- 「Broadcast Repeater ユーティリティ (dbns12)」921 ページ

## Host 接続パラメーターを使用した SQL Anywhere データベースサーバーの検出

Host 接続パラメーターを使って TCP/IP 経由でデータベースサーバーを検出しようとしている SQL Anywhere クライアントは、次の手順を使ってサーバーを検出します。

1. 手順 1：データベースサーバーアドレスキャッシュ (sasrv.ini) を確認する.
2. 手順 2：Host 接続パラメーターで指定されたアドレスに TCP/IP 接続を試みる.
3. 手順 3：UDP データベースサーバー検索要求を送信する.

データベースサーバーが見つからない場合は、サーバーが見つからなかったというエラーが返されます。別のエラーが発生した場合は、接続はそれ以上の手順を行うことなく失敗します。「データベースサーバーが見つかりません。」『エラーメッセージ』を参照してください。

クライアントは、データベースサーバーへの接続に成功した場合、データベースに接続しようとします。「データベースの検出」91 ページを参照してください。

**手順 1：データベースサーバーアドレスキャッシュ (*sasrv.ini*) を確認する**　接続文字列に Host 接続パラメーターは含まれているものの ServerName は含まれていない場合、SQL Anywhere クライアントは *sasrv.ini* ファイルを使用できません。 手順 2：Host 接続パラメーターで指定されたアドレスに TCP/IP 接続を試みる85 ページを参照してください。

クライアントは、ServerName 接続パラメーターによって指定されたデータベースサーバー名に一致するエントリをクライアント内の *sasrv.ini* ファイルで確認します。

一致するキャッシュエントリが見つからない場合、クライアントは直接 TCP/IP 接続を試みます。手順 2：Host 接続パラメーターで指定されたアドレスに TCP/IP 接続を試みる85 ページを参照してください。

キャッシュエントリが ServerName 接続パラメーターと、Host 接続パラメーターで指定されたアドレスの両方に一致する場合、クライアントはキャッシュに示されている IP アドレスへの TCP/IP 接続を試みます。

○クライアントは、この IP アドレスを使ってデータベースサーバーに接続する場合、データベースサーバーの名前と ServerName で指定された値を比較します。

データベースサーバー名が一致する場合、接続は成功します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

データベースサーバー名が一致しない場合、クライアントはその *sasrv.ini* ファイルからエントリを削除し、直接 TCP/IP 接続を試みます。手順 2：Host 接続パラメーターで指定されたアドレスに TCP/IP 接続を試みる85 ページを参照してください。

○クライアントは、接続できない場合、その *sasrv.ini* ファイルからエントリを削除し、直接 TCP/IP 接続を試みます。手順 2：Host 接続パラメーターで指定されたアドレスに TCP/IP 接続を試みる85 ページを参照してください。

> **注意**
> UNIX 上のデータベースサーバーアドレスキャッシュファイル名は *.sasrv.ini.* です。「高速の接続のための sasrv.ini でのデータベースサーバーアドレス情報のキャッシング」91 ページを参照してください。

**手順 1：例**　ホスト名が kangaroo、IP アドレスが 10.25.13.5 のコンピューターがあるとします。joey という名前のデータベースサーバーがこのコンピューターのポート 49152 で実行中です。

接続文字列 Host=kangaroo;ServerName=joey に対して、クライアントはサーバー名 joey と一致するエントリをその *sasrv.ini* ファイルで検索します。キャッシュ内の IP アドレスは 10.25.13.5:49152 です。クライアントは指定された Host 名のアドレスと 10.25.13.5 を比較します。アドレスが一致するので、クライアントは TCP/IP 経由で 10.25.13.5:49152 に接続してから、接続したサーバーの名前が joey であることを確認します。接続は成功です。

**手順 2：Host 接続パラメーターで指定されたアドレスに TCP/IP 接続を試みる**　複数のアドレスが指定された場合、クライアントは、Host 接続パラメーターに指定された順序で各アドレスに接続しようとします。クライアントは、接続に成功するまで、またはすべてのアドレスを試してみるまで接続を試みます。

アドレスにポートが含まれる場合、クライアントは、そのアドレスと指定されたポート番号への TCP/IP 接続を試みます。ポートが指定されない場合、クライアントはデフォルトポートである 2638 で指定されたアドレスの TCP/IP 接続を試みます。

○クライアントは、データベースサーバーに接続し、ServerName が指定された場合、データベースサーバーの名前と ServerName で指定された値を比較します。

データベースサーバー名が一致する場合、接続は成功です。クライアントはその *sasrv.ini* ファイルを更新します。

データベースサーバー名が一致しない場合、クライアントは UDP サーバー検索要求を使って接続しようとします。手順 3：UDP データベースサーバー検索要求を送信する86 ページを参照してください。

○クライアントがデータベースサーバーに接続して ServerName が指定されない場合、接続は成功です。

○クライアントが接続できず、ServerName が指定された場合、クライアントは UDP サーバー検索要求を使って接続しようとします。手順 3：UDP データベースサーバー検索要求を送信する86 ページを参照してください。

○クライアントが接続できず、ServerName が指定されない場合、接続は失敗します。

**手順 2：例**　ホスト名が kangaroo、IP アドレスが 10.25.13.5 のコンピューターがあるとします。joey という名前のデータベースサーバーは、このコンピューターのポート 49152 で実行中です。クライアントがこのサーバーに接続されたのはこれが初めてなので、キャッシュされていないアドレスが *sasrv.ini* ファイルに存在します。

接続文字列 Host=kangaroo:49152;ServerName=joey に対して、クライアントは TCP/IP 経由で 10.25.13.5:49152 に接続してから、接続したサーバーの名前が joey であることを確認します。接続は成功なので、クライアントはその *sasrv.ini* ファイルを更新します。

**手順 3：UDP データベースサーバー検索要求を送信する**　SQL Anywhere クライアントは、接続文字列に ServerName 接続パラメーターが含まれており、Host 値がポートを指定しないときにだけ、UDP データベースサーバー検索要求を送信します。UDP サーバー検索要求は、指定されたアドレスでポート 2638 に送信されます。

デフォルトでは、同一のコンピューター上のすべてのデータベースサーバー (Mac OS X を実行しているコンピューターを除く) は、ポート 2638 で UDP パケットを受信します。Mac OS X では、1 つのサーバーだけがポート 2638 で UDP パケットを受信できます。データベースサーバーが UDP サーバー検索要求を受信すると、データベースサーバーは、その名前と UDP 要求で指定された名前を比較します。名前が一致する場合、データベースサーバーはその IP アドレスとポート番号を含む UDP 応答パケットを送信し戻します。一致するデータベースサーバーだけが応答パケットを送信します。

デフォルトでは、クライアントは UDP 応答を 5 秒間待ちます。応答が受信されない場合、このプロセスがタイムアウトするまで UDP パケットが毎秒再送信されます。

クライアントが UDP 応答パケットを受信すると、クライアントはパケットで指定されたアドレスとパケットにポートへの TCP/IP 接続を試みます。

クライアントがデータベースサーバーに接続できる場合、接続は成功です。クライアントはその *sasrv.ini* ファイルを更新します。

**注意**
UDP パケットは次のデータベースサーバーを検出できません。

○ポート 2638 からの受信を行わない Mac OS X 上で実行中のデータベースサーバー。「ServerPort (PORT) プロトコルオプション」396 ページを参照してください。

○-sb 0 オプションが指定されているデータベースサーバー。「-sb dbeng12/dbsrv12 サーバーオプション」270 ページを参照してください。

○ファイアウォール、ルーター、またはゲートウェイによってブロックされる UDP 要求の送信元であるデータベースサーバー。「ファイアウォール経由の接続」101 ページを参照してください。

これらすべての場合で、Host 接続パラメーターにデータベースサーバーのポートを指定することが、クライアントに接続の成功を許可します。

**手順 3：例**　ホスト名が kangaroo、IP アドレスが 10.25.13.5 のコンピューターがあるとします。joey という名前のデータベースサーバーは、このコンピューターのポート 49152 で実行中です。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

クライアントがこのサーバーに接続されたのはこれが初めてなので、キャッシュされていないアドレスが *sasrv.ini* ファイルに存在します。

接続文字列 Host=kangaroo;ServerName=joey において、クライアントは TCP/IP 経由で 10.25.13.5:2638 に接続しようとして、失敗しました (手順 2)。接続文字列にポート番号が指定されていないので、クライアントは UDP サーバー検索要求をアドレス 10.25.13.5:2638 に送信します。データベースサーバー joey は、UDP アドレス 10.25.13.5.2638 で受信を待っており、アドレス 10.25.13.5:49152 で応答します。クライアントは TCP/IP 経由で 10.25.13.5:49152 に接続します。接続は成功するので、クライアントはその *sasrv.ini* ファイルを更新します。

## CommLinks=TCPIP 接続パラメーターを使用した SQL Anywhere データベースーサーバーの検出

**注意**
CommLinks (LINKS) 接続パラメーターは、HOST または ServerPort (PORT) 以外の TCP/IP プロトコルオプションを指定する必要がある場合のみ使用することをおすすめします。これ以外の場合は、Host 接続パラメーターを使用します。「Host 接続パラメーター」346 ページを参照してください。

接続文字列に CommLinks と Host の両方は指定できません。

Host 接続パラメーターは、HOST プロトコルオプションとは別のものです。HOST プロトコルオプションは、CommLinks 接続パラメーターによって使用されます。 「Host (IP) プロトコルオプション (クライアント側のみ)」381 ページを参照してください。

CommLinks 接続パラメーターを使って TCP/IP 経由でサーバーを検出しようとしている SQL Anywhere クライアントは、次の手順を使ってサーバーを検出します。

1. 手順 1：データベースサーバーアドレスキャッシュ (sasrv.ini) を確認する87 ページ
2. 手順 2：サーバー名について LDAP サーバーを確認する88 ページ
3. 手順 3：HOST プロトコルオプションで指定されたアドレスに TCP/IP 接続を試みる 89 ページ
4. 手順 4：UDP データベースサーバー検索要求を送信する89 ページ

データベースサーバーが見つからない場合は、サーバーが見つからなかったというエラーが返されます。別のエラーが発生した場合は、接続はそれ以上の手順を行うことなく失敗します。「データベースサーバーが見つかりません。」『エラーメッセージ』を参照してください。

クライアントは、データベースサーバーへの接続に成功した場合、データベースに接続しようとします。「データベースの検出」91 ページを参照してください。

**手順 1 : データベースサーバーアドレスキャッシュ (*sasrv.ini*) を確認する** 接続文字列が HOST プロトコルオプションを含んでいるものの、ServerName 接続パラメーターは含んでいない場合、LocalOnly プロトコルオプションが YES に設定されるか、DoBroadcast プロトコルオプションが

NONE に設定されるので、SQL Anywhere クライアントは直接 TCP/IP 接続を試みます。つまり、SQL Anywhere クライアントは、データベースサーバーアドレスキャッシュまたは LDAP サーバーを使用しません。手順 3：HOST プロトコルオプションで指定されたアドレスに TCP/IP 接続を試みる89 ページを参照してください。

クライアントは、ServerName 接続パラメーターで指定されたデータベースサーバー名に一致するエントリをクライアント内の *sasrv.ini* ファイルで確認します。接続文字列が HOST プロトコルオプションも含む場合、キャッシュエントリは指定されたホストアドレスにも一致する必要があります。

一致するキャッシュエントリが見つからない場合、クライアントは LDAP を使ってデータベースサーバーを検出しようとします。手順 2：サーバー名について LDAP サーバーを確認する88 ページを参照してください。

一致するキャッシュエントリが見つかった場合、クライアントはキャッシュでリストされている IP アドレスに TCP/IP 接続を試みます。

○クライアントがこの IP アドレスを使ってデータベースサーバーに接続し、VerifyServer プロトコルオプションが No に設定される場合、接続は成功します。「VerifyServerName (VERIFY) プロトコルオプション (クライアント側のみ)」402 ページを参照してください。

○クライアントは、この IP アドレスを使ってデータベースサーバーに接続し、VerifyServer が Yes (デフォルト) に設定される場合、クライアントはデータベースサーバーの名前と ServerName で指定された値を比較します。データベースサーバー名が一致する場合、接続は成功します。データベースサーバー名が一致しない場合、クライアントは、その *sasrv.ini* ファイルからエントリを削除し、LDAP を使ってデータベースサーバーを検出しようとします。手順 2：サーバー名について LDAP サーバーを確認する88 ページを参照してください。

○クライアントは、接続できない場合、その *sasrv.ini* ファイルからエントリを削除し、LDAP を使ってデータベースサーバーを検出しようとします。手順 2：サーバー名について LDAP サーバーを確認する88 ページを参照してください。

**注意**
UNIX 上のデータベースサーバーアドレスキャッシュファイル名は *.sasrv.ini.* です。「高速の接続のための sasrv.ini でのデータベースサーバーアドレス情報のキャッシング」91 ページを参照してください。

**手順 1：例**　ホスト名が kangaroo、IP アドレスが 10.25.13.5 のコンピューターがあるとします。joey という名前のデータベースサーバーがこのコンピューターのポート 49152 で実行中です。

接続文字列 CommLinks=TCPIP(Host=kangaroo);ServerName=joey に対して、クライアントはサーバー名 joey と一致するエントリをその *sasrv.ini* ファイルで検索します。キャッシュ内の IP アドレスは 10.25.13.5:49152 です。クライアントは指定された Host 名のアドレスと 10.25.13.5 を比較します。アドレスが一致するので、クライアントは TCP/IP 経由で 10.25.13.5:49152 に接続してから、接続したサーバーの名前が joey であることを確認します。接続は成功です。

**手順 2：サーバー名について LDAP サーバーを確認する**　SQL Anywhere クライアントは、SQL Anywhere クライアントとデータベースサーバーのコンピューターが LDAP サーバーを使用するように設定されているときに、ServerName 接続パラメーターに一致するエントリの LDAP サーバーをクエリします。「LDAP サーバーを使用した接続」103 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**手順 3 : HOST プロトコルオプションで指定されたアドレスに TCP/IP 接続を試みる**　HOST プロトコルオプションが指定されない場合、SQL Anywhere クライアントは UDP サーバー検索要求を使ってサーバーを検出しようとします。手順 4 : UDP データベースサーバー検索要求を送信する 89 ページを参照してください。

SQL Anywhere クライアントは、TCP/IP を経由して HOST プロトコルオプションに指定されたアドレスに接続しようとします。複数のアドレスが指定された場合、クライアントは、HOST プロトコルオプションに指定された順序で各アドレスに接続しようとします。クライアントは、接続に成功するまで、またはすべてのアドレスを試してみるまで接続を試みます。

接続文字列がポートを含むとき (HOST プロトコルオプションまたは ServerPort 接続プロパティのいずれか)、クライアントは指定されたアドレスとポート番号への TCP/IP 接続を試みます。ポートが指定されない場合、クライアントはデフォルトポートである 2638 で指定されたアドレスの TCP/IP 接続を試みます。

○クライアントが指定された IP アドレスを使ってデータベースサーバーに接続し、VerifyServer プロトコルオプションが No に設定される場合、接続は成功します。「VerifyServerName (VERIFY) プロトコルオプション (クライアント側のみ)」402 ページを参照してください。

○クライアントは、指定されたアドレスを使ってデータベースサーバーに接続した場合、データベースサーバーの名前と ServerName で指定された値を比較します。データベースサーバー名が一致する場合、接続は成功し、クライアントはその *sasrv.ini* ファイルを更新します。データベースサーバー名が一致しない場合、クライアントは UDP サーバー検索要求を使って接続しようとします。手順 4 : UDP データベースサーバー検索要求を送信する89 ページを参照してください。

○クライアントが接続できず、ServerName が指定された場合、クライアントは UDP サーバー検索要求を使って接続しようとします。手順 4 : UDP データベースサーバー検索要求を送信する 89 ページを参照してください。

○クライアントが接続できず、ServerName が指定されない場合、接続は失敗します。

**手順 3 : 例**　ホスト名が kangaroo、IP アドレスが 10.25.13.5 のコンピューターがあるとします。joey という名前のデータベースサーバーは、このコンピューターのポート 49152 で実行中です。クライアントがこのサーバーに接続されたのはこれが初めてなので、キャッシュされていないアドレスが *sasrv.ini* ファイルに存在します。

接続文字列 CommLinks=TCPIP(Host=kangaroo:49152);ServerName=joey に対して、クライアントは TCP/IP 経由で 10.25.13.5:49152 に接続してから、接続したサーバーの名前が joey であることを確認します。接続は成功なので、クライアントはその *sasrv.ini* ファイルを更新します。

**手順 4 : UDP データベースサーバー検索要求を送信する**　SQL Anywhere クライアントは、接続文字列に ServerName 接続パラメーターが含まれており、DoBroadcast プロトコルオプションは DIRECT または ALL に設定されているときにだけ、UDP データベースサーバー検出要求を送信します。

○DoBroadcast=DIRECT または HOST プロトコルオプションが指定されたとき、クライアントは UDP サーバー検索パケットを指定されたアドレスに送信します。ポートが指定されたとき (HOST プロトコルオプションまたは ServerPort プロトコルオプションのいずれかで)、UDP

サーバー検索要求は指定されたポートに送信されます。それ以外の場合、要求はデフォルトポート 2638 に送信されます。

○DoBroadcast=ALL が指定され、HOST プロトコルオプションが指定されない場合、クライアントは、IP アドレスのそれぞれに対するサブネットのブロードキャストアドレスを判断します。ServerPort プロトコルオプションが指定された場合、クライアントは UDP ブロードキャストサーバー検索パケットを指定のポートに送信します。それ以外の場合、UDP ブロードキャストパケットはデフォルトのポート 2638 に送信されます。

データベースサーバーが起動されたときに -sb 0 オプションが指定されていない限り、すべてのデータベースサーバーがポート 2638 で UDP パケットを受信します。データベースサーバーが UDP サーバー検索要求を受信すると、データベースサーバーはその名前と要求パケットで指定された名前を比較します。名前が一致する場合、データベースサーバーはその IP アドレスとポート番号を含む UDP 応答パケットを送信し戻します。一致するデータベースサーバーだけが応答パケットを送信します。

デフォルトでは、クライアントは UDP 応答を 5 秒間待ちます。応答が受信されない場合、このプロセスがタイムアウトするまで UDP パケットが毎秒再送信されます。

クライアントが UDP 応答パケットを受信すると、クライアントはパケットで指定されたアドレスとパケットにポートへの TCP/IP 接続を試みます。

クライアントがデータベースサーバーに接続できる場合、接続は成功です。クライアントはその *sasrv.ini* ファイルを更新します。

**注意**
UDP パケットは次のデータベースサーバーを検出できません。

○異なるサブネット上に存在するデータベースサーバー。UDP サーバー検出ブロードキャストパケットは、Broadcast Repeater ユーティリティが使用されない限り、データベースサーバーを検索できません。「Broadcast Repeater ユーティリティ (dbns12)」921 ページを参照してください。

○ポート 2638 からの受信を行わない Mac OS X 上で実行中のデータベースサーバー。「ServerPort (PORT) プロトコルオプション」396 ページを参照してください。

○-sb 0 オプションが指定されているデータベースサーバー。「-sb dbeng12/dbsrv12 サーバーオプション」270 ページを参照してください。

○ファイアウォール、ルーター、またはゲートウェイによってブロックされる UDP 要求の送信元であるデータベースサーバー。「ファイアウォール経由の接続」101 ページを参照してください。

いずれの場合にも、HOST プロトコルオプションでデータベースサーバーのアドレス (ポート番号を含む) を指定することによって、クライアントは正常に接続できるようになります。

**手順 4：例**　ホスト名が kangaroo、IP アドレスが 10.25.13.5 のコンピューターがあるとします。joey という名前のデータベースサーバーは、このコンピューターのポート 49152 で実行中です。クライアントがこのサーバーに接続されたのはこれが初めてなので、キャッシュされていないア

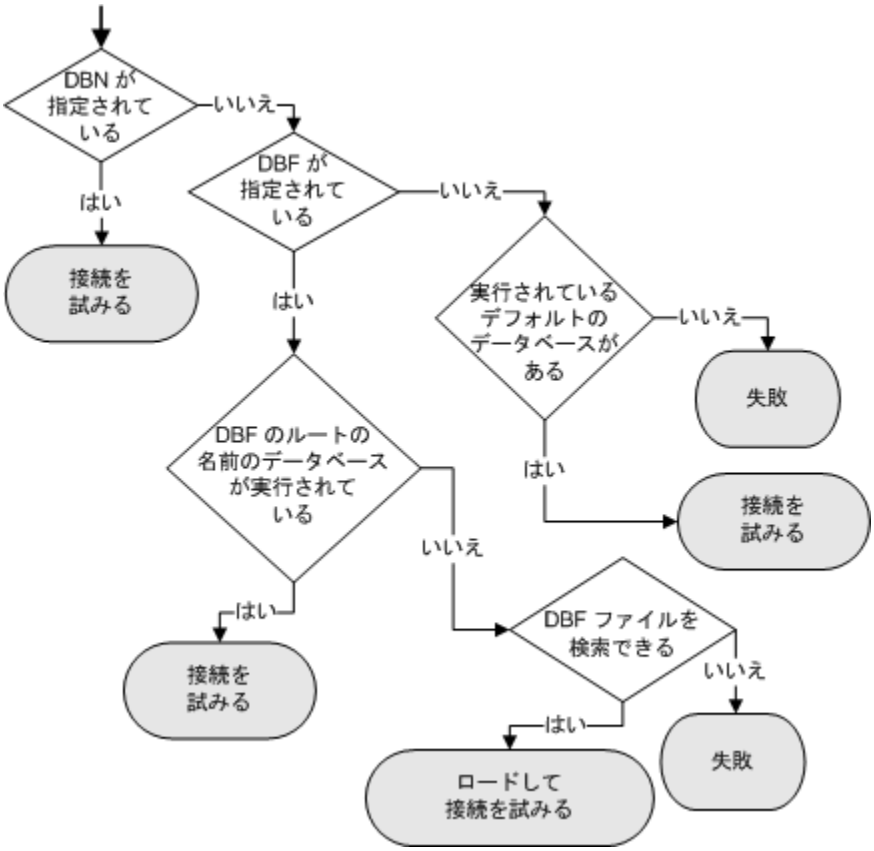

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

ドレスが *sasrv.ini* ファイルに存在します。接続は kangaroo と同じサブネット上にあるコンピューターから試行され、このサブネットのブロードキャストアドレスは 10.25.13.255 です。

接続文字列 CommLinks=TCPIP;ServerName=joey の場合、クライアントは UDP サーバー検索ブロードキャスト要求をブロードキャストアドレス 10.25.13.255:2638 に送信します。データベースサーバー joey は、UDP アドレス 10.25.13.5.2638 で受信を待っており、アドレス 10.25.13.5:49152 で応答します。クライアントは TCP/IP 経由で 10.25.13.5:49152 に接続します。接続は成功するので、クライアントはその *sasrv.ini* ファイルを更新します。

## データベースの検出

SQL Anywhere は、サーバーの検出に成功すると、データベースの検出に移ります。




## 高速の接続のための *sasrv.ini* でのデータベースサーバーアドレス情報のキャッシング

Host または CommLinks 接続パラメーターを使用するとき、特定の名前のデータベースサーバーを検索するために複数の TCP/IP 接続を試行するか、大規模ネットワーク経由でブロードキャス

トすることは、時間がかかる可能性があります。データベースサーバーアドレスのキャッシュは、後の接続に使用されるように次の情報を *sasrv.ini* ファイルに保存することによって、ネットワーク接続を高速化します。

●データベースサーバー名

●データベースサーバーの IP アドレス

キャッシュファイルには一連のセクションがあり、それぞれが次の形式になっています。

```
[Server name]
LINK=TCPIP
Address=IP_address:port
```

**注意**
それぞれのデータベースサーバーの名前がユニークであることは、重要です。異なるデータベースサーバーに同じ名前を付けると、識別の問題を引き起こす可能性があります。

### データベースサーバーアドレスキャッシュ (*sasrv.ini*) が使用される方法

*sasrv.ini* ファイルが使用される方法の詳細については、「Host 接続パラメーターを使用した SQL Anywhere データベースサーバーの検出」84 ページと「CommLinks=TCPIP 接続パラメーターを使用した SQL Anywhere データベースーサーバーの検出」87 ページを参照してください。

### *sasrv.ini* ファイルのロケーション

*sasrv.ini* ファイルのデフォルトロケーションは次のとおりです。

| オペレーティングシステム | *sasrv.ini* のデフォルトロケーション |
|---|---|
| Windows | *%ALLUSERSPROFILE%¥Application Data¥SQL Anywhere 12* |
| Windows Mobile | 実行されている実行プログラムと同じディレクトリ |
| UNIX | *$HOME/.sqlanywhere12*<br><br>UNIX では、データベースサーバーアドレスキャッシュファイルの名前は *.sasrv.ini.* です。 |

### 参照


- 「高速の接続のための sasrv.ini でのデータベースサーバーアドレス情報のキャッシング」91 ページ
- 「Host 接続パラメーターを使用した SQL Anywhere データベースサーバーの検出」84 ページ
- 「CommLinks=TCPIP 接続パラメーターを使用した SQL Anywhere データベースーサーバーの検出」87 ページ


 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## データベースサーバーを見つけられるかどうかのテスト (dbping)

接続のトラブルシューティングには dbping ユーティリティを使用します。

-c オプション付きで dbping ユーティリティを使用すると、接続文字列パラメーターをテストできます。デフォルトでは、dbping はデータベースには接続せず、サーバーに接続するだけであり、サーバーを起動しません。dbping ユーティリティに -d オプションを使用すると、必要に応じて、データベースに接続して、データベースサーバーを起動します。

### 例

たとえば、demo12 という名前のパーソナルサーバーがサンプルデータベース (コマンド dbeng12 -n demo12 "%SQLANYSAMP12%¥demo.db" で実行可能) を実行しているとします。ローカルコンピューター上で demo12 という名前のデータベースサーバーが実行中であり、demo という名前のデータベースが実行されている場合、次の文字列は ping が成功したことを示すメッセージを返します。

```
dbping -d -c "Server=demo12;DBN=demo;UID=DBA;PWD=sql"
```

次のコマンドは、デフォルトポート 2638 の Waterloo という名前のコンピューターでデータベースサーバーを検出できるかどうかをテストします。

```
dbping -c "Host=Waterloo"
```

次のコマンドは、デフォルトデータベースサーバーが現在のコンピューター上で実行中であるかどうかをテストします。

```
dbping
```

### 参照


- 「Ping ユーティリティ (dbping)」994 ページ
- 「Embedded SQL とネットワーク接続パフォーマンスのテスト (dbping)」93 ページ


## Embedded SQL とネットワーク接続パフォーマンスのテスト (dbping)

Embedded SQL 接続とネットワークのパフォーマンスに関する情報を取得するには、ping ユーティリティ (dbping) を使用し、-s または -st オプションを指定します。次の統計値が収集されます。

| 統計情報 | 説明 |
| --- | --- |
| DBLib の接続と切断 | DBLib の接続および切断を 1 度実行する時間。ODBC などの他のインターフェイスを使用した接続および切断は、接続の完了に必要な要求が多いため、一般的にパフォーマンスは DBLib より低くなります。 |

| 統計情報 | 説明 |
|---|---|
| 簡単な要求の往復時間 | クライアントからサーバーに要求を送信するのにかかる時間とサーバーからクライアントに応答を送信するのにかかる時間との和。往復時間は平均遅延時間の 2 倍になります。 |
| 送信スループット | dbping からデータベースサーバーへの反復ごとに 100 KB のデータを転送する場合のスループット。 |
| 受信スループット | データベースサーバーから dbping への反復ごとに 100 KB のデータを転送する場合のスループット。 |

往復時間が長く、スループットが高いネットワークの場合、往復時間が長いために、レポートされるスループットはネットワークの実際のスループットより低くなります。通信圧縮によりパフォーマンスが向上するかどうかを確認するには、dbping -s を使用すると便利です。パフォーマンス統計は概算値であり、クライアントコンピューターとサーバーコンピューターの両方がアイドル状態である方が統計値の精度は高くなります。通信圧縮を使用すると、送信されるデータは元のサイズの約 25% に圧縮されます。

dbping -s コマンドからの出力例を次に示します。次の dbping コマンドを実行しました。

```
dbping -s -c "UID=DBA;PWD=sql;Host=localhost;Server=demo12"

SQL Anywhere サーバー Ping ユーティリティバージョン 12.0.1.3149
SQL Anywhere 12.0.1.3149 サーバー "sampleserver" とデータベース
"sample" にアドレス 10.25.107.108 で接続しました。

パフォーマンス統計              数値            合計時間       平均
---------------------------   -------------   ----------   -----------
DBLib との接続と切断            200 回          10 ミリ秒    < 1 ミリ秒
単純要求のラウンドトリップ      10900 要求      1024 ミリ秒  <1 ミリ秒
送信スループット                86800 KB        1024 ミリ秒  84765 KB／秒
受信スループット                84300 KB        1024 ミリ秒  82324 KB／秒
データベースへの ping が成功しました。
```

**参照**

- 「Ping ユーティリティ (dbping)」994 ページ
- 「データベースサーバーを見つけられるかどうかのテスト (dbping)」93 ページ

## 互換性のあるプロトコルを使用していることの確認

クライアントとデータベースサーバーが同じプロトコルを使用していることを確認してください。

- **共有メモリプロトコル**　クライアントとデータベースサーバーが同じコンピューター上にある場合、共有メモリ (デフォルトプロトコル) をおすすめします。

- **TCP/IP プロトコル**　クライアントとデータベースサーバーが異なるコンピューター上にある場合は、TCP/IP を使用する必要があります。ネットワークサーバーはデフォルトで TCP/IP 接続を許可しているのに対して、パーソナルサーバーは許可していません。パーソナルサーバーは、-x tcpip オプションを指定することによって、ローカルの (ただし、リモートは不可)

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

TCP/IP 接続を受け付けることができます。クライアントは、Host または CommLinks 接続パラメーターを使用して、データベースサーバーへの TCP/IP 接続を確立します。

**参照**

- 「-x dbeng12/dbsrv12 サーバーオプション」289 ページ
- 「Host 接続パラメーター」346 ページ
- 「CommLinks (LINKS) 接続パラメーター」327 ページ

## TCP/IP パフォーマンス

TCP/IP 要求パケットサイズを大きくすると、クエリの応答時間を短縮できます。特に、クライアントとデータベースサーバープロセスの間で大量のデータを転送するクエリでは大変有効です。パケットサイズを設定するには、-p dbeng12/dbsrv12 サーバーオプションを使用するか、接続プロファイルに CommBufferSize (CBSIZE) 接続パラメーターを設定します。

TCP/IP のチューニングが特定の状況で有益であるかどうかを判断するには、運用環境で調整を行う前に、アプリケーションを使用しているネットワークでパフォーマンス分析を行います。

**参照**

- 「-p dbeng12/dbsrv12 サーバーオプション」262 ページ
- 「CommBufferSize (CBSIZE) 接続パラメーター」326 ページ

## 頻度が高い問題のチェックリスト

次のリストに、頻度が高い問題のいくつかとその解決策を示します。

データベースまたはデータベースサーバーへの接続に関するトラブルシューティングについては、「接続のトラブルシューティング」76 ページと「データベースサーバー起動時のトラブルシューティング」1211 ページを参照してください。

接続しようとしたときにデータベースサーバーが見つからないというメッセージを受信した場合は、クライアントがネットワークでデータベースサーバーを検索できていません。「データベースサーバーが見つかりません。」『エラーメッセージ』を参照してください。

次のような問題がないか調べてください。

- 接続文字列にサーバー名を指定します。

> **注意**
> コンピューターは、組み込みの SQL Anywhere サーバーを含むソフトウェアを実行している可能性があり、その場合、気づかずに複数のサーバーが実行されているかもしれません。常にサーバー名を指定する (ServerName=*server-name*) ことをおすすめします。

- リモートデータベースサーバーに対しては、Host 接続パラメーターを指定し、TCP/IP ホスト名またはデータベースサーバーを実行中のコンピューターのアドレスを提供することをおす

すめします。指定したアドレスが正しいことを確認します。サーバーがデフォルトの 2638 以外のポートを使用している場合、Host=*tcpip-address*:*port-number* を指定する必要があります。「Host 接続パラメーター」346 ページを参照してください。

詳細 CommLinks 接続パラメーターを使用している場合、データベースサーバーの検索に使用されるパケットが、データベースサーバーに到達しない可能性があります。Host、ServerPort、または DoBroadcast 以外の TCP/IP プロトコルオプションを使用する必要がある場合を除いて、Host 接続パラメーターを使用することをおすすめします。HOST プロトコルオプションを指定せずに CommLinks 接続パラメーターを使用している場合、UDP ブロードキャストまたは要求は、現在のサブネットに限定されるか、ルーター、ゲートウェイ、またはファイアウォールによってブロックされます。

- クライアントとサーバーの間にファイアウォールがあり、接続が妨害されている可能性がある。「ファイアウォール経由の接続」101 ページを参照してください。

- パーソナルサーバーは同じコンピューターからの接続だけを受け入れる。クライアントとサーバーが異なるコンピューター上にある場合は、ネットワークサーバーを使用する必要があります。

- ネットワークドライバーが正しくインストールされていないか、またはネットワーク配線が正しくない。クライアントコンピューターがサーバーコンピューターと通信できるかどうかを確認するには、ping ユーティリティを使用します。

- jConnect または Sybase Open Client 経由で接続している場合、サーバーが TCP/IP プロトコルを使用する必要がある。

- 管理ツール、ODBC、または OLE DB の **[接続]** ウィンドウを使用しており、ローカルコンピューター上でデータベースサーバーに接続しようとしている場合、**[接続]** ウィンドウの **[このコンピューターで稼働しているデータベースに接続]** をクリックします。ローカルコンピューター以外のコンピューターで稼働しているデータベースサーバーに接続する場合は、**[別のコンピューターで稼働しているデータベースに接続]** をクリックできます。

**参照**

- 「ネットワークプロトコルオプション」369 ページ

# 通信プロトコル

データベースサーバーは、さまざまなプロトコルを介して通信します。SQL Anywhere には、サポートされているプロトコルの動作をカスタマイズするための多くのオプションがあります。

HTTP 接続と HTTPS 接続の詳細については、「HTTP Web サーバーとしての SQL Anywhere」『SQL Anywhere サーバー プログラミング』と「Web クライアントを使用した Web サービスへのアクセス」『SQL Anywhere サーバー プログラミング』を参照してください。

TDS の使用の詳細については、「インターフェイスライブラリ通信プロトコル」『SQL Anywhere 12 紹介』を参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

# 通信プロトコルオプションの考慮事項

クライアントアプリケーションとデータベースサーバー間の通信では、通信プロトコルが必要です。SQL Anywhere は、ネットワーク通信用と同一コンピューター通信用の通信プロトコルセットをサポートします。

### パーソナルデータベースサーバーで使用できるプロトコル

デフォルトでは、パーソナルデータベースサーバーは共有メモリ接続のみを許可します。TCP/IP プロトコルも起動するには、-x オプションを使用します。「-x dbeng12/dbsrv12 サーバーオプション」289 ページを参照してください。

- **共有メモリ**　このプロトコルは、同一コンピューター通信で使用され、常にすべてのプラットフォームで使用可能です。

  同一コンピューター通信では、共有メモリの方が TCP/IP よりもパフォーマンスが高くなります。

  UNIX での共有メモリ接続の保護については、SATMP 環境変数をユニークなディレクトリに設定 (UNIX)1381 ページを参照してください。

- **TCP/IP**　このプロトコルは、パーソナルサーバー上の同一コンピューター通信に対してのみ使用可能です。また、TDS クライアント (Sybase Open Client または jConnect JDBC ドライバー) を使用している場合は、TCP/IP を使用する必要があります。すべてのプラットフォームで使用できます。

  TDS クライアントの詳細については、「Open Server としての SQL Anywhere」1435 ページを参照してください。

### ネットワークデータベースサーバー (dbsrv12) で使用できるプロトコル

デフォルトでは、ネットワークデータベースサーバーは共有メモリプロトコルと TCP/IP プロトコルを起動します。TCP/IP プロトコルオプションを提供することも、-x オプションを使用して TCP/IP プロトコルを無効にすることもできます。「-x dbeng12/dbsrv12 サーバーオプション」289 ページを参照してください。

- **共有メモリ**　このプロトコルは、同一コンピューター通信で使用され、常にすべてのプラットフォームで使用可能です。UNIX での共有メモリ接続の保護については、SATMP 環境変数をユニークなディレクトリに設定 (UNIX)1381 ページを参照してください。

- **TCP/IP**　このプロトコルは、主に異なるコンピューター間の通信に使用されます。TDS クライアント (Sybase Open Client または jConnect JDBC ドライバー) を使用している場合は、TCP/IP を使用する必要があります。すべてのプラットフォームで使用できます。TDS クライアントの詳細については、「Open Server としての SQL Anywhere」1435 ページを参照してください。

### SQL Anywhere クライアントからの接続

- **共有メモリ**　共有メモリは、デフォルトの接続プロトコルです。共有メモリ経由で接続する場合は、クライアントの接続文字列には Host と CommLinks のどちらの接続パラメーターも指定しないでください。

- **TCP/IP**　TCP/IP を経由して接続する場合は、Host または CommLinks 接続パラメーターのいずれかを指定します。Host 接続パラメーターを指定することをおすすめします。「Host 接続パラメーター」346 ページと「CommLinks (LINKS) 接続パラメーター」327 ページを参照してください。

**参照**


- 「TCP/IP プロトコル」99 ページ
- 「ネットワークプロトコルオプション」369 ページ
- 「接続パラメーター」320 ページ


## 共有メモリのセッションとターミナルについての考慮事項

### 共有メモリ、サービス、ターミナルサービス (Windows Vista 以降)

データベースサーバーとクライアントをサービスとしてまたはターミナルサービス内で実行する場合：

- クライアントは、共有メモリ経由でサービスとして動作する任意のデータベースサーバーに接続できます。
- クライアントは、クライアントと同じセッションで実行中であるデータベースサーバーに共有メモリ経由で接続できます。共有メモリ接続を行うときに、クライアントは最初にクライアントと同じセッションで指定された名前のデータベースサーバーを探します。何も見つからない場合、クライアントは指定された名前のサービスを探します。
- 1 つのターミナルセッションで実行中のクライアントは、別のターミナルセッションで実行中のデータベースサーバーに対して共有メモリ経由で接続することはできません。代わりに、TCP/IP を使用します。
- サービスとして実行するクライアントは、1 つのターミナルセッションで実行中のデータベースサーバーに対して共有メモリ経由で接続することはできません。代わりに、TCP/IP を使用します。
- 現在のセッションで同じ名前のデータベースサーバーがすでに実行中の場合、または同じ名前で実行中のサービスがある場合、データベースサーバーは、起動**できません**。
- 別のセッションで同じ名前で実行中のデータベースサーバーがある場合は、同じ名前で実行中のサービスがなく、データベースサーバーがどちらも TCP/IP を使用しない限り、データベースサーバーは起動**できます**。

### 共有メモリ、サービス、ターミナルサービス (Windows XP および 2003)

データベースサーバーとクライアントをサービスとしてまたはターミナルサービス内で実行する場合：

- Windows XP では、サービスとプライマリデスクトップはどちらも同じセッション内で実行されます。この設定をプライマリセッションと呼びます。
- クライアントは、クライアントと同じセッションで実行中であるか、またはプライマリセッションで実行中であるデータベースサーバーに共有メモリ経由で接続できます。共有メモリ


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

接続を行うときに、クライアントは最初にクライアントと同じセッション内、次にプライマリセッション内で、指定された名前のデータベースサーバーを探します。

- プライマリセッション内で実行中のクライアントは、プライマリ以外のセッションで実行中のデータベースサーバーに対して共有メモリ経由で接続することはできません。プライマリ以外のセッション内で実行中のクライアントは、プライマリ以外の別のセッションで実行中のデータベースサーバーに対して共有メモリ経由で接続することはできません。代わりに、TCP/IP を使用します。

- 現在のセッション内またはプライマリセッション内で同じ名前でデータベースサーバーがすでに実行中である場合、データベースサーバーは起動**できません**。(プライマリセッションの)デスクトップ上で実行中のデータベースサーバーは、同じ名前のサービスが起動するのを妨げる可能性があります。ただし、通常、サービスは、システムが起動するときに起動するので、この問題が発生するのはまれです。

- プライマリ以外の別のセッションで共有メモリ経由で実行中の同じ名前のデータベースサーバーがある場合であっても、プライマリセッションで同じ名前で実行中のデータベースサーバーが存在しない限り、データベースサーバーは共有メモリを使って起動**できます**。

- 他の任意のセッション (プライマリまたはプライマリ以外) で TCP/IP 経由で実行中の同じ名前のデータベースサーバーが存在する場合、データベースサーバーは TCP/IP を使っては起動**できません**。

**参照**

- 「通信プロトコルオプションの考慮事項」97 ページ

# TCP/IP プロトコル

SQL Anywhere は、異なるコンピューター上で実行中のデータベースにクライアントを接続するために TCP/IP を使用します。

TCP/IP プロトコルを使用するときは、トランスポートレイヤーセキュリティと ECC または RSA 暗号化テクノロジを使用して、クライアント／サーバー通信を安全化できます。「TCP/IP を使用したクライアント／サーバー通信の暗号化」100 ページを参照してください。

## UDP

UDP は、トランスポートレイヤーのプロトコルです。これは IP の最上部のプロトコルです。SQL Anywhere は、初期のサーバー名解決を行うのに UDP を使用し、その後、TCP を使用して接続と通信を行います。

クライアントのブロードキャストへの応答としてデータベースサーバーから送信される UDP パケットには機密情報が含まれません。これらのパケットに含まれるデータは次のとおりです。

- データベースサーバー名
- ポート番号
- データベースサーバーのバージョン
- データベースサーバーで実行されているデータベースの名前

-dh データベースオプションを使用すると、UDP ブロードキャスト応答に対してデータベース名を難読化することができます。

-sb サーバーオプションを指定すると、UDP リスナーを無効にできます。

### Windows での TCP/IP の使用

すべての Windows プラットフォーム上のデータベースサーバーの TCP/IP 実装では、Winsock 2.2 を使用します。Windows Mobile 上のクライアントは、Winsock 1.1 standard を使用します。

### 参照


- 「-dh dbeng12/dbsrv12 データベースオプション」309 ページ
- 「-sb dbeng12/dbsrv12 サーバーオプション」270 ページ
- 「-x dbeng12/dbsrv12 サーバーオプション」289 ページ
- 「Host 接続パラメーター」346 ページ
- 「Broadcast (BCAST) プロトコルオプション」372 ページ
- 「Host (IP) プロトコルオプション (クライアント側のみ)」381 ページ
- 「MyIP (ME) プロトコルオプション」394 ページ


## TCP/IP を使用したクライアント／サーバー通信の暗号化

デフォルトでは通信パケットが暗号化されないため、セキュリティに関して潜在的な危険があります。単純暗号化またはトランスポートレイヤーセキュリティを使用して、TCP/IP を介したクライアントアプリケーションとデータベースサーバー間の通信を安全化できます。トランスポートレイヤーセキュリティにより、サーバー認証、ECC または RSA 暗号化テクノロジを使用した強力な暗号化、およびデータ整合性を保護するその他の機能が提供されます。

### 参照


- 「トランスポートレイヤーセキュリティ」1410 ページ


## SQL Anywhere での IPv6 サポート

IPv6 対応のコンピューターでは、ネットワークデータベースサーバーは、デフォルトで IPv4 と IPv6 のすべてのアドレスを受信します。IPv6 は、Windows CE ではサポートされません。

通常、IPv6 を使用するようデータベースサーバーの開始行を変更する必要はありません。IP アドレスの指定が必要な場合は、データベースサーバーライブラリとクライアントライブラリは、どちらも IPv4 アドレスと IPv6 アドレスを受け付けます。たとえば、コンピューターで複数のネットワークカードを使用できる場合、2 つの IPv4 アドレスと 2 つの IPv6 アドレスが存在することがあります。IPv6 アドレスの場合は、アドレスを角カッコまたはカッコで囲む必要があります。データベースサーバーが受信する IPv6 アドレスを 1 つに制限する場合は、アドレスを次のフォーマットで指定できます。

```
dbsrv12 -x tcpip(MyIP=fd77:55f:5a64:52a:202:5445:5245:444f) ...
```

同様に、クライアントアプリケーションでサーバーの IP アドレスを指定する必要がある場合は、接続文字列または ODBC データソースに次のフォーマットでアドレスを指定できます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
...HOST=fd77:55f:5a64::444f;...
```

各インターフェイスにインターフェイス識別子が与えられており、IPv6 アドレスの末尾に示されます。たとえば、*ipconfig.exe* にアドレス fd77:55f:5a64::444f がリストされた場合、インターフェイス識別子は 7 です。IPv6 アドレスを Windows プラットフォームに指定する場合は、インターフェイス識別子を使用する必要があります。UNIX では、インターフェイス識別子とインターフェイス名のどちらでも指定できます (インターフェイス名は、ifconfig によってレポートされるインターフェイス識別子の名前です)。たとえば、IPv6 アドレスが fd77:55f:5a64::444f; である場合、インターフェイス名は eth1 です。Linux (カーネル 2.6.13 以降) で IPv6 アドレスを指定する場合は、インターフェイス識別子が必要です。この要件は次によって指定された値に影響します。

- 「Host 接続パラメーター」
- 「Broadcast (BCAST) プロトコルオプション」
- 「Host (IP) プロトコルオプション (クライアント側のみ)」
- 「MyIP (ME) プロトコルオプション」

### 例

*ipconfig.exe* に 2 つのインターフェイスがリストされており、片方の識別子が 1 でもう片方の識別子が 2 だととします。インターフェイス番号 2 によって使用されているネットワーク上のデータベースサーバーを検索している場合、クライアントライブラリに対してそのインターフェイスにだけブロードキャストするよう指示できます。

```
LINKS=tcpip(BROADCAST=ff02::1%2)
```

IPv6 リンクローカルのマルチキャストアドレスは ff02::1 です。

## ファイアウォール経由の接続

クライアントアプリケーションがファイアウォールの片側にあり、データベースサーバーがもう一方の側にある場合は、接続が制限されます。ファイアウォールソフトウェアは、ネットワークポートに従って、ネットワークパケットをフィルタリングする可能性があります。また通常は、UDP パケットはファイアウォールを通過できません。

通常、接続は、Host 接続パラメーターを使用して適切に設定されたファイアウォールを経由し、データベースサーバーのアドレスとポートを提供することで実行できます。データベースサーバーがデフォルトポート 2638 を使用している場合、ポートは必要ありません。

次の接続文字列は、ポート 2020 を使用するアドレス serverhost のコンピューター上で実行中の myserver という名前のデータベースサーバーに接続します。Host 接続パラメーターが TCP/IP アドレスとポートを指定しているので、UDP パケットは使用されません。

```
Server=myserver;Host=serverhost:2020
```

### 特定のクライアントポートのみを許可するファイアウォール

特定のクライアントポートのみを許可するようにファイアウォールを設定する必要がある場合、Host 接続パラメーターではなく、CommLinks(LINKS) 接続パラメーターを使用する必要がありま

す。次の TCP/IP プロトコルオプションは、CommLinks 接続パラメーターを使用するときに必要です。

- **Host**　このプロトコルオプションには、データベースサーバーを実行しているホスト名を設定します。IP と短縮してもかまいません。
- **ServerPort**　データベースサーバーがデフォルトポート 2638 を使用していない場合、使用しているポートを指定します。Port と短縮してもかまいません。
- **ClientPort**　このプロトコルオプションには、使用するクライアントアプリケーションで有効な範囲の値を設定します。CPORT と短縮してもかまいません。
- **DoBroadcast=NONE**　サーバーに接続するときに UDP が使用されないようにするには、このプロトコルオプションを設定します。DOBROAD と短縮してもかまいません。

ファイアウォールは、SQL Anywhere データベースサーバーのアドレスとすべての SQL Anywhere クライアントのアドレスとの間の TCP/IP トラフィックを許可するように設定します。SQL Anywhere データベースサーバーのアドレスは、SQL Anywhere サーバーを実行中のコンピューターの IP アドレス (HOST プロトコルオプション) と SQL Anywhere データベースサーバーの IP ポート番号 (ServerPort プロトコルオプション、デフォルトは 2638) です。

クライアントポートが再び利用されるまでに数分のタイムアウトがあるため、各クライアントコンピューターからの同時接続の最大数よりも多いクライアントポート数の範囲を使用します。ClientPort プロトコルオプションで指定されるクライアントの範囲は、ファイアウォールで許可される範囲と一致する必要があります。

**参照**


- 「CommLinks (LINKS) 接続パラメーター」327 ページ
- 「ClientPort (CPORT) プロトコルオプション (クライアント側のみ)」377 ページ
- 「ServerPort (PORT) プロトコルオプション」396 ページ
- 「Host (IP) プロトコルオプション (クライアント側のみ)」381 ページ
- 「DoBroadcast (DOBROAD) プロトコルオプション」379 ページ


**例**

次の接続文字列は、クライアントアプリケーションをポート 5050 ～ 5060 に制限します。また、サーバーポート 2020 を使用するアドレス myhost のコンピューターで実行されているデータベースサーバー myeng に接続します。DoBroadcast プロトコルオプションを NONE に設定しているため、UDP ブロードキャストは実行されません。

```
Server=myeng;LINKS=tcpip(ClientPort=5050-5060;HOST=myhost;PORT=2020;DoBroadcast=NONE)
```

## ダイヤルアップネットワーク接続 (CommLinks 接続パラメーター)

ダイヤルアップネットワーク接続経由で接続するには、CommLinks 接続パラメーターを使用する必要があります。

クライアント側では、以下のプロトコルオプションを指定してください。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

- **Host**　Host (IP) プロトコルオプションを使用して、データベースサーバーのホスト名または IP アドレスを指定します。「Host (IP) プロトコルオプション (クライアント側のみ)」381 ページを参照してください。

- **DoBroadcast**　Host (IP) プロトコルオプションを指定した場合は、データベースサーバーでブロードキャスト検索を行う必要はありません。このため、ダイレクトブロードキャストを使用してください。「DoBroadcast (DOBROAD) プロトコルオプション」379 ページを参照してください。

- **MyIP**　クライアント側では、**MyIP=NONE** に設定してください。「MyIP (ME) プロトコルオプション」394 ページを参照してください。

- **TIMEOUT**　サーバーを検索する間のクライアントの待機時間を長くするには、TIMEOUT (TO) プロトコルオプションを設定します。「Timeout (TO) プロトコルオプション」400 ページを参照してください。

**例**

次に示すのは、ダイヤルアップネットワーク接続経由で接続するために使用する一般的な CommLinks (LINKS) 接続パラメーターの例です。

```
LINKS=tcpip(MyIP=NONE;DoBroadcast=DIRECT;HOST=server-ip); TIMEOUT=15)
```

## LDAP サーバーを使用した接続

Windows (Windows Mobile を除く) または UNIX プラットフォーム上で動作している場合、中央 LDAP サーバーを指定して企業内の全データベースサーバーを追跡できます。データベースサーバー自体を LDAP サーバーに登録する場合、クライアントは LDAP サーバーに問い合わせてデータベースサーバーを検索できます。この場合、サーバーが WAN 上、LAN 上、またはファイアウォールの外側にあっても検索できます。LDAP は、Host 接続パラメーターまたは Host プロトコルオプションでデータベースサーバーアドレスを指定するクライアントからは無視されます。

サーバー列挙ユーティリティ (dblocate) も LDAP サーバーを使用してそのようなサーバーを検索できます。

LDAP は TCP/IP とともに使用し、ネットワークデータベースサーバーでのみ使用されます。

### IBM AIX での SQL Anywhere と LDAP サーバーの使用

**♦ /usr/lib にリンクを作成するには、次の手順に従います。**

IBM AIX 6 で SQL Anywhere を使用するには、*/usr/lib* にリンクを作成するか、LDAP ライブラリが含まれるディレクトリが LIBPATH に含まれていることを確認し、LDAP のシステムライブラリが見つかるようにします。

1. root ユーザーで次のコマンドを実行します。

```
cd /usr/lib
ln -s /opt/IBM/ldap/V6.1/lib64/libibmldap.a libibmldap64.a
ln -s /opt/IBM/ldap/V6.1/lib/libibmldap.a
```

2. root ユーザーで次のコマンドを実行して */usr/lib* にリンクを作成します。

```
cd /usr/lib
ln -s /opt/IBM/ldap/V6.1/lib64/libibmldap.a libibmldap64.a
ln -s /opt/IBM/ldap/V6.1/lib/libibmldap.a
```

3. LDAP ライブラリのあるディレクトリが LIBPATH に含まれていることを確認します。

   たとえば、64 ビットのライブラリの場合は、次のように入力します。

```
export LIBPATH=/opt/IBM/ldap/V6.1/lib64:$LIBPATH
```

   たとえば、32 ビットのライブラリの場合は、次のように入力します。

```
export LIBPATH=/opt/IBM/ldap/V6.1/lib:$LIBPATH
```

**参照**

- 「LDAP プロトコルオプション」387 ページ
- 「サーバー列挙ユーティリティ (dblocate)」999 ページ

## *saldap.ini* ファイルの設定

LDAP サーバーを使用して接続するには、LDAP サーバーの検索方法と接続方法に関する情報を含むファイルをデータベースサーバーコンピューターと各クライアントコンピューターに作成してください。デフォルトで、このファイル名は *saldap.ini* ですが、この名前は変更できます。*saldap.ini* ファイルの名前を変更する場合、ファイル名を指定するために LDAP プロトコルオプションを使用する必要があります。このファイルがない場合、LDAP のサポートは無効になっています。

ファイル難読化ユーティリティ (dbfhide) を使用して、単純暗号化によって *saldap.ini* ファイルの内容を難読化することができます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。

このファイルは、LDAP プロトコルオプションに完全なパスを指定していない場合は SQL Anywhere 実行プログラム (たとえば、Windows の場合は *%SQLANY12%¥bin32*) と同じディレクトリ置く必要があります。このファイルの内容は、次のフォーマットになります。

```
[LDAP]
server=computer-running-LDAP-server
port=port-number-of-LDAP-server
basedn=Base-DN
authdn=Authentication-DN
password=password-for-authdn
search_timeout=age-of-timestamps-to-be-ignored
update_timeout=frequency-of-timestamp-updates
read_authdn=read-only-authentication-domain-name
read_password=password-for-authdn
```

**server**　LDAP サーバーを実行中のコンピューターの名前または IP アドレス。この値は UNIX で必要です。このエントリが Windows にない場合、Windows がローカルのドメインコントローラーで動作中の LDAP サーバーを検索します。

**port**　LDAP サーバーで使用されるポート番号。デフォルトは 389 です。

**basedn**　SQL Anywhere エントリが格納されているサブツリーのドメイン名。この値はデフォルトでツリーのルートになります。

**authdn**　認証ドメイン名。ドメイン名は、LDAP ディレクトリにある既存のユーザーオブジェクトである必要があります。このディレクトリには basedn への書き込みアクセスがあります。このパラメーターはデータベースサーバーで必要ですが、クライアントでは無視されます。

**password**　authdn のパスワード。このパラメーターはデータベースサーバーで必要ですが、クライアントでは無視されます。

**search_timeout**　タイムスタンプがクライアントまたはサーバー列挙ユーティリティ (dblocate) で無視されるタイムスタンプの経過時間。値 0 はこのオプションを無効にして、すべてのエントリが現在のものと見なされます。デフォルト値は 600 秒 (10 分) です。

**update_timeout**　LDAP ディレクトリ内のタイムスタンプの更新頻度。値として 0 を指定するとこのオプションは無効になり、データベースサーバーはタイムスタンプを更新しません。デフォルト値は 120 秒 (2 分) です。

**read_authdn**　読み取り専用の認証ドメイン名。ドメイン名は、LDAP ディレクトリにある既存のユーザーオブジェクトを指定します。このディレクトリには basedn への読み取りアクセスがあります。このパラメーターが必要なのは、検索の実行前に LDAP サーバーで非匿名バインドが必要な場合だけです。たとえば、Active Directory が LDAP サーバーとして使用されている場合、通常はこのフィールドが必要です。このパラメーターが指定されていない場合、バインドは匿名になります。

**read_password**　authdn のパスワード。クライアントにこのパラメーターが必要なのは、read_authdn パラメーターが指定されている場合だけです。

### 接続の確立方法

データベースサーバーは、起動時に、LDAP 内で同じ名前を持つ既存のエントリがないかどうかを確認します。一致するエントリが見つかった場合、LDAP のロケーションエントリと起動しようとしているデータベースサーバーとが一致した場合、または LDAP エントリのタイムスタンプフィールドが search_timeout パラメーターで指定されている時間よりも長い場合に、エントリは置き換えられます。

あるデータベースサーバーが起動しようとするときに同じ名前で実行中の別のデータベースサーバーが存在する場合、データベースは起動できません。

LDAP のエントリを確実に最新のものにするために、データベースサーバーは LDAP エントリ内のタイムスタンプフィールドを 2 分ごとに更新します。エントリのタイムスタンプが 10 分以上古い場合、クライアントは LDAP エントリを無視します。これらの設定はいずれも変更可能です。

クライアントでは、クライアントが UDP ブロードキャストを実行する前に LDAP ディレクトリが検索されるので、データベースサーバーが見つかると UDP ブロードキャストは送信されません。LDAP 検索は非常に高速なので、失敗しても認識できるほどの遅延は発生しません。

### LDAP サーバーとサーバー列挙ユーティリティ (dblocate)

サーバー列挙ユーティリティ (dblocate) も LDAP を使用します。LDAP にリストされているすべてのデータベースサーバーが、返されるデータベースサーバーのリストに追加されます。これにより、サーバー列挙ユーティリティ (dblocate) が、たとえばブロードキャストが到達しなかったデータベースサーバーなど、通常どおりに返されなかったサーバーをリストします。10 分以上古いタイムスタンプを持つエントリは含まれません。

#### 参照

● 「サーバー列挙ユーティリティ (dblocate)」 999 ページ

#### 例

次に、*saldap.ini* ファイルの例を示します。

```
[LDAP]
server=ldapserver
basedn=dc=iAnywhere,dc=com
authdn=cn=SAServer,ou=iAnywhereASA,dc=iAnywhere,dc=com
password=secret
```

エントリが iAnywhereASA と呼ばれる basedn のサブツリーに保管されます。このエントリは SQL Anywhere が LDAP を使用できるようになる前に作成します。サブツリーを作成するには、LDAPADD ユーティリティを使用します。次のような情報を提供します。

```
dn: ou=iAnywhereASA,basedn
objectClass: organizationalUnit
objectClass: top
ou: iAnywhereASA
```

## 通信の圧縮設定

1 つまたはすべての接続について圧縮機能を有効にして、パケット圧縮時の最小サイズを設定すると、SQL Anywhere のパフォーマンスを向上できる場合があります。そのため、圧縮スレッショルドを変更することが有益かどうかを判断するために、ネットワーク上でアプリケーションとのパフォーマンス分析を実行する必要があります。

圧縮機能を有効にすると、データパケットに格納される情報量が増大し、特定のデータセットの送信に必要なパケット数が減少します。パケット数を減らすと、データを高速で送信できます。

圧縮の有効化が特定の状況で有益であるかどうかを判断するには、通信の圧縮を運用環境で使用する前に、アプリケーションを使用しているネットワークでパフォーマンス分析を行います。

次の項目によって圧縮のパフォーマンスを調整できます。

**圧縮の有効化**　次のような状況では、1 つ (またはすべて) の接続で圧縮を有効化すると、SQL Anywhere のパフォーマンスが大幅に向上する可能性があります。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

○一部の無線ネットワーク、モデム、シリアルリンク、WAN などの低速ネットワークで使用するとき。

○圧縮機能が組み込まれている低速ネットワークで SQL Anywhere 暗号化を使用するとき。これは、パケットが圧縮されてから暗号化されるためです。

ただし、圧縮の有効化は、パフォーマンス低下の原因になる可能性もあります。次に例を示します。

○通信の圧縮には、より多くのメモリと CPU が使用される。このため、特に LAN やその他の高速ネットワークを使用する場合に、パフォーマンスが低下することがあります。

○ほとんどのモデムと一部の低速ネットワークに、すでに圧縮機能が組み込まれている。この場合、SQL Anywhere で通信圧縮を行っても、データの暗号化を同時に実行しないかぎり、パフォーマンスはほとんど向上しません。

圧縮を有効または無効にする方法については、「Compress (COMP) 接続パラメーター」329 ページと「-pc dbsrv12 サーバーオプション」263 ページを参照してください。

### 圧縮のスレッショルドの変更

**注意**
ほとんどのネットワークでは、圧縮のスレッショルドを変更する必要はありません。まれではありますが、SQL Anywhere のパフォーマンスは、圧縮のスレッショルドを調整することでも向上できる可能性があります。

圧縮スレッショルドの変更が特定の状況で有益であるかどうかを判断するには、運用環境で調整を行う前に、アプリケーションを使用しているネットワークでパフォーマンス分析を行います。

圧縮が有効な場合、パケットは、各々のサイズに応じて圧縮するかどうかを決定します。SQL Anywhere では、圧縮のスレッショルドよりも小さいパケットは、通信の圧縮が有効な場合でも圧縮されません。

パケットの圧縮には CPU 時間が必要なので、小さなパケットを圧縮しようとすることで、実際にはパフォーマンスが低下することもあります。

一般に、圧縮のスレッショルド値を小さくすると、非常に低速なネットワークではパフォーマンスが向上し、値を大きくすると CPU 使用率の減少によってパフォーマンスが向上する場合があります。ただし、圧縮のスレッショルド値を小さくするとクライアントとサーバーの両方で CPU 使用率が増加するので、パフォーマンス分析を行って、圧縮のスレッショルドを変更することでパフォーマンスが向上するかどうかを判断してください。

**♦ SQL Anywhere 圧縮設定を調整するには、次の手順に従います。**

1. 通信の圧縮を有効にします。

   高度に圧縮可能なデータを大きなパケットサイズで大量にデータ転送することで、最高の圧縮率を得ることができます。「Compress (COMP) 接続パラメーター」329 ページと「-pc dbsrv12 サーバーオプション」263 ページを参照してください。

2. CompressionThreshold 設定を調整します。

   圧縮スレッショルドの値を小さくすると、非常に低速なネットワークではパフォーマンスが向上し、値を大きくすると CPU 使用率の減少によってパフォーマンスが向上する場合があります。

   CompressionThreshold (COMPTH) 接続パラメーターの調整については、「CompressionThreshold (COMPTH) 接続パラメーター」330 ページと「-pt dbsrv12 サーバーオプション」264 ページを参照してください。

**参照**

- 「パフォーマンスモニターの統計値」『SQL Anywhere サーバー SQL の使用法』
- 「sa_conn_compression_info システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』

# チュートリアル：SQL Anywhere データベースの作成

このチュートリアルでは、SQL Anywhere サンプルデータベースの Products、SalesOrderItems、SalesOrders、Customers の各テーブルを使用してモデル化した単純なデータベースを Sybase Central で作成する方法について説明します。

## レッスン 1：データベースファイルの作成

次の手順を使用して、データベースを格納するデータベースファイルを作成します。データベースファイルには、システムテーブルと、すべてのデータベースに共通するその他のシステムオブジェクトが格納されます。

**♦ 新しいデータベースファイルを作成するには、次の手順に従います (Sybase Central の場合)。**

1. **[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[Sybase Central]** をクリックします。

2. **[ツール]** » **[SQL Anywhere 12]** » **[データベースの作成]** をクリックします。

3. **[ようこそ]** ページで、**[次へ]** をクリックします。

4. **[このコンピューターにデータベースを作成]** をクリックし、**[次へ]** をクリックします。

5. **[メインデータベースファイルを保存]** フィールドに、*c:¥temp¥mysample.db* と入力します。

   テンポラリフォルダーとして *c:¥temp* 以外のディレクトリを使用する場合は、適切なパスを指定します。

6. **[完了]** をクリックします。

7. **[閉じる]** をクリックします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

8. 「レッスン 2 ：データベースへのテーブルの追加」109 ページに進みます。

**参照**

- 「SQL Anywhere サンプルデータベース」『SQL Anywhere 12 紹介』
- 「データベースの作成 (SQL の場合)」118 ページ
- 「データベースの作成 (コマンドラインの場合)」119 ページ
- 「レッスン 2 ：データベースへのテーブルの追加」109 ページ
- 「レッスン 3：カラムへの NOT NULL 制約の設定」111 ページ
- 「レッスン 4：外部キーの作成」111 ページ

# レッスン 2 ：データベースへのテーブルの追加

次の手順を使用して、Products という名前のテーブルを作成します。

**♦ テーブルを作成するには、次の手順に従います。**

1. Sybase Central の右ウィンドウ枠で、**[テーブル]** をダブルクリックします。
2. **[テーブル]** を右クリックし、**[新規]** » **[テーブル]** をクリックします。
3. **[新しいテーブルの名前を指定してください。]** フィールドに、**Products** と入力します。
4. **[完了]** をクリックします。

   データベースサーバーによって、デフォルト設定を使用してテーブルが作成され、右ウィンドウ枠に **[カラム]** タブが表示されます。新しいカラムの **[名前]** フィールドが選択され、新しいカラムの名前を指定するよう求められます。

5. 新しいカラムの名前として、**ProductID** と入力します。

   このカラムがテーブル内の最初のカラムなので、テーブルのプライマリキーであることを示す **[プライマリキー]** が選択されます。

   テーブルを作成する場合、カラムを作成し、各 **[プライマリキー]** カラムのチェックマークをオンにすることで、複数のカラムから成るプライマリキーを作成できます。「プライマリキー」『SQL Anywhere サーバー SQL の使用法』を参照してください。

6. **[データ型]** リストで、**[integer]** をクリックします。
7. 省略記号 (ピリオド 3 つ) ボタンをクリックします。
8. **[値]** タブをクリックし、**[デフォルト値]** » **[システム定義]** » **[autoincrement]** をクリックします。

   AUTOINCREMENT 値は、テーブルにローが追加されるごとに増加します。これにより、カラムにはユニークな値が設定されます。これはプライマリキーの要件です。「プライマリキー」『SQL Anywhere サーバー SQL の使用法』を参照してください。

9. **[OK]** をクリックします。

10. **[ファイル]** » **[新規]** » **[カラム]** をクリックします。

11. 次のフィールドを完成させます。

- **[名前]** フィールドに、**ProductName** と入力します。
- **[データ型]** リストで、**[char]** をクリックします。
- **[サイズ]** リストで、**[15]** をクリックします。

12. 次のテーブルをデータベースに追加します。

- **Customers テーブル**　テーブル **Customers** を追加し、テーブル内に以下のカラムを作成します。
  - **CustomerID**　各顧客の ID 番号です。**[プライマリキー]** が選択されていることを確認し、**[データ型]** を **[integer]** に、**[デフォルト値]** を **[autoincrement]** にそれぞれ設定します。
  - **CompanyName**　各会社の名前です。**[データ型]** を **[char]** に設定し、最大長を **35** 文字にします。
- **SalesOrders テーブル**　テーブル **SalesOrders** を追加し、テーブル内に以下のカラムを作成します。
  - **SalesOrderID**　各注文書の ID 番号です。**[データ型]** を **[integer]** に設定し、**[プライマリキー]** が選択されていることを確認します。**[デフォルト値]** に **[autoincrement]** を設定します。
  - **OrderDate**　注文日です。**[データ型]** を **[date]** に設定します。
  - **CustomerID**　発注した顧客の ID 番号です。**[データ型]** を **[integer]** に設定します。
- **SalesOrderItems テーブル**　テーブル **SalesOrderItems** を追加し、テーブル内に以下のカラムを作成します。
  - **SalesOrderItemsID**　項目を含む注文書の ID 番号です。**[データ型]** を **[integer]** に設定し、**[プライマリキー]** が選択されていることを確認します。
  - **LineID**　各注文書の ID 番号です。**[データ型]** を **[integer]** に設定し、**[プライマリキー]** が選択されていることを確認します。

    > **注意**
    > **[プライマリキー]** は、SalesOrderItemsID と LineID の両方に設定されているため、この 2 つのカラムの連結値でテーブルのプライマリキーが構成されます。

  - **ProductID**　受注製品の ID 番号です。**[データ型]** を **[integer]** に設定します。

13. **[ファイル]** » **[保存]** をクリックします。

14. 「レッスン 3：カラムへの NOT NULL 制約の設定」111 ページに進みます。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**参照**

- 「テーブルの作成」『SQL Anywhere サーバー SQL の使用法』
- 「レッスン 1：データベースファイルの作成」108 ページ
- 「レッスン 3：カラムへの NOT NULL 制約の設定」111 ページ
- 「レッスン 4：外部キーの作成」111 ページ
- 「SQL Anywhere 12 プラグインからの ER 図の表示」835 ページ

# レッスン 3：カラムへの NOT NULL 制約の設定

次の手順を使用して、NOT NULL 制約をカラムに追加します。

**♦ カラムに制約を追加する、またはカラムから制約を削除するには、次の手順に従います。**

1. Sybase Central の左ウィンドウ枠で、**[テーブル]** をダブルクリックします。
2. **[Products]** をクリックし、右ウィンドウ枠で **[カラム]** タブをクリックします。
3. **[ProductName]** カラムをクリックします。
4. **[ファイル]** メニューで、**[プロパティ]** をクリックします。
5. **[制約]** タブをクリックし、**[NULL 値を禁止]** をクリックします。

   デフォルトではカラムに NULL 値が許可されますが、NULL 値を許可する明確な理由がないかぎり、カラムには NOT NULL を宣言してください。「NULL 特別値」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

6. **[OK]** をクリックします。

   この制約により、**[Products]** テーブルに追加されたローごとに、**[ProductName]** カラムに値が必要になります。

7. **[ファイル]** メニューで、**[保存]** をクリックします。
8. 「レッスン 4：外部キーの作成」111 ページに進みます。

**参照**

- 「レッスン 1：データベースファイルの作成」108 ページ
- 「レッスン 2 ：データベースへのテーブルの追加」109 ページ
- 「レッスン 4：外部キーの作成」111 ページ

# レッスン 4：外部キーの作成

次の手順を使用して、外部キーによりテーブル間の関係を作成します。

**♦ 外部キーを作成するには、次の手順に従います。**

1. Sybase Central の左ウィンドウ枠で、**[テーブル]** をダブルクリックします。

2. 左ウィンドウ枠で、**[SalesOrdersItems]** テーブルをクリックして選択します。
3. 右ウィンドウ枠で、**[制約]** タブをクリックします。
4. **[ファイル]** » **[新規]** » **[外部キー]** をクリックします。
5. **[この外部キーが参照するテーブルを指定してください。]** リストで、**[Products]** テーブルをクリックします。
6. **[新しい外部キーの名前を指定してください。]** フィールドに、**ProductIDkey** と入力します。
7. **[次へ]** をクリックし、**[この外部キーが参照する主キー制約または一意性制約を指定してください。]** で、**[プライマリキー]** をクリックします。
8. **[外部カラム]** リストで、**[SalesOrderItemsID]** をクリックします。
9. **[完了]** をクリックします。

**参照**


- 「外部キーの作成 (Sybase Central の場合)」『SQL Anywhere サーバー SQL の使用法』
- 「外部キーの作成 (SQL の場合)」『SQL Anywhere サーバー SQL の使用法』
- 「チュートリアル：サンプルデータベースへの接続」1 ページ
- 「レッスン 1：データベースファイルの作成」108 ページ
- 「レッスン 2 ：データベースへのテーブルの追加」109 ページ
- 「レッスン 3：カラムへの NOT NULL 制約の設定」111 ページ


# データベースファイルのタイプ

各データベースには、次のような関連ファイルがあります。

- **データベースファイル**　このファイルは、データベースの内容全体を保持します。1 つのファイルで 1 つのデータベースを構成する方法と、最高 12 の DB 領域を追加する方法があります。DB 領域は、同じデータベースの一部を入れる追加ファイルです。データベースファイルと DB 領域のロケーションを選択します。通常、データベースファイルの拡張子は *.db* です。

- **トランザクションログ**　このファイルはデータベースの変更記録を格納するものであり、障害が発生した場合に、データベースの情報の同期とリカバリを行うのに必要です。通常、トランザクションログの拡張子は *.log* です。「トランザクションログ」127 ページを参照してください。

- **テンポラリファイル**　データベースサーバーは、SQL Anywhere で、ソートや UNION 処理など、操作用キャッシュで使用できる領域よりも多くの領域が必要な場合、データベースセッション中にテンポラリファイルを使用します。こうした領域が必要な場合、データベースサーバーは、通常はその領域を集中的に使用します。データベース全体のパフォーマンスは、テンポラリファイルを含むデバイスの速度に大きく依存します。データベースが停止すると、データベースサーバーはたとえ実行中であっても、テンポラリファイルを破棄します。サーバーが生成した名前に拡張子 *.tmp* を付けたものが、テンポラリファイル名です。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

テンポラリファイルのロケーションは、データベースサーバーの起動時に -dt サーバーオプションを使用して指定できます。テンポラリファイルのロケーションを指定せずにデータベースサーバーを起動した場合は、以下の環境変数がこの順序のとおりにチェックされます。

○ SATMP 環境変数
○ TMP 環境変数
○ TMPDIR 環境変数
○ TEMP 環境変数

これらの環境変数がいずれも定義されていない場合、テンポラリファイルは、Windows オペレーティングシステムの場合は現在のディレクトリ、UNIX の場合は */tmp* ディレクトリに作成されます。

テンポラリファイルは、データベースサーバーが作成、管理、削除します。テンポラリファイル用に使用できる、十分な空き領域があることだけ確認してください。テンポラリファイルに使用できる領域に関する情報は、sa_disk_free_space プロシージャーを使用すると取得できます。「sa_disk_free_space システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

- **事前定義の DB 領域ファイル**　これらのファイルには、データと、データベースで使用されるその他のファイルが格納されます。「事前定義の DB 領域」120 ページを参照してください。
- **DB 領域ファイル**　データベースファイルに加えて、データを複数の個別ファイルに分散できます。「追加の DB 領域の考慮事項」121 ページを参照してください。
- **トランザクションログミラーファイル**　データの保全のため、トランザクションログのミラーコピーを作成できます。通常、このファイルの拡張子は *.mlg* です。「トランザクションログミラー」128 ページを参照してください。

**参照**

- 「異なるファイルの異なるデバイスへの配置」『SQL Anywhere サーバー SQL の使用法』

# SQL Anywhere データベースの作成

SQL Anywhere データベースを作成するには、まず、データベースに格納されるテーブル (エンティティ)、各テーブルのカラム (属性)、テーブル間の関係 (キーと制約) を定義します。

新しいデータベースには概念データベースモデル (CDM) を作成することをおすすめします。Sybase PowerDesigner などの CDM アプリケーションを使用して、データベース設計を作成し、検証できます。これらのツールで作成された SQL 文は、通常 ODBC インターフェイスを通してデータベースサーバーに送信されます。このようなツールを使えば、テーブルの作成、パーミッションの割り当てなどを行う SQL 文を構築する必要はありません。

データベースの設計と CDM の作成の詳細については、http://infocenter.sybase.com/help/index.jsp で入手できる Sybase PowerDesigner のマニュアルを参照してください。

テーブルやビューなどのデータベースオブジェクトの詳細については、「テーブル、ビュー、インデックス」『SQL Anywhere サーバー SQL の使用法』を参照してください。

また、Sybase Central、Interactive SQL、またはコマンドラインを使用して、SQL Anywhere データベースを作成または「初期化」できます。データベースを作成したら、そのデータベースに接続して、テーブルやその他のオブジェクトを追加できます。

### トランザクションログ

データベースを作成するには、トランザクションログを配置する場所を決定しなければなりません。このログには、データベースへ加えられた変更内容が、変更された順に格納されます。データベースファイルでメディア障害が発生した場合、トランザクションログはデータベースのリカバリに重要な役割を果たします。また、トランザクションログを使用すると、より効率的に作業できます。デフォルトでは、トランザクションログはデータベースファイルと同じディレクトリに配置されますが、この設定での運用はおすすめしません。

トランザクションログの配置の詳細については、「トランザクションログ」127 ページを参照してください。

### データベースファイルの互換性

SQL Anywhere データベースはオペレーティングシステムファイルです。このデータベースは、他のファイルと同様に別のロケーションにコピーできます。

ファイルシステムのファイルサイズの制限事項や大きなファイルに対する SQL Anywhere のサポートが適用される場合以外は、すべてのオペレーティングシステム間でデータベースファイルには互換性があります。「SQL Anywhere のサイズと数の制限」811 ページを参照してください。

オペレーティングシステム間の互換は、データベースファイルをコピーすれば可能です。同様に、パーソナルデータベースサーバーで作成されたデータベースは、ネットワークデータベースサーバーで使用できます。SQL Anywhere データベースサーバーは、旧バージョンのソフトウェアで作成されたデータベースを管理できますが、旧バージョンのサーバーは新バージョンのデータベースを管理できません。

### 参照


- 「SQL Anywhere 12 プラグインからの ER 図の表示」835 ページ
- PowerDesigner Physical Data Model『SQL Anywhere 12 紹介』


## オブジェクト名の考慮事項

データベースオブジェクト名には予約語を使用しないでください。

カラム名に英字、数字、アンダースコア以外のものが含まれていたり、カラム名が英字で始まっていなかったり、キーワードと同じであったりする場合、カラム名を二重引用符 (" ") で囲みます。

### 参照


- 「予約語」『SQL Anywhere サーバー SQL リファレンス』


 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

# カラムデータ型の考慮事項

SQL Anywhere では、次のデータ型を使用できます。

- バイナリデータ型
- 文字データ型
- 日付／時刻データ型
- decimal データ型
- ドメイン (ユーザー定義データ型)
- 浮動小数点データ型
- integer データ型
- 空間データ型

CHAR、VARCHAR、LONG VARCHAR、NCHAR、BINARY、VARBINARY などの文字列やバイナリ文字列のデータ型を使用して、イメージ、ワープロ文書、音声ファイルなどのラージオブジェクトを格納できます。

**参照**

- 「BLOB の考慮事項」116 ページ
- 「SQL データ型」『SQL Anywhere サーバー SQL リファレンス』

## NULL と NOT NULL カラム

カラム値がローに対して必須である場合、そのカラムを NOT NULL になるように定義します。そうしないと、カラムには NULL 値が許可され、値を表示しません。SQL Anywhere のデフォルトは NULL 値を許容しますが、NULL 値を許容する正当な理由がないかぎり、カラムには NOT NULL を明示的に宣言してください。

SQL Anywhere サンプルデータベースには Departments という名前のテーブルがあり、そのテーブルには DepartmentID、DepartmentName、DepartmentHeadID カラムがあります。カラムの定義は次のとおりです。

| カラム | データ型 | サイズ | NULL／NOT NULL | 制約 |
|---|---|---|---|---|
| DepartmentID | integer | – | NOT NULL | プライマリキー |
| DepartmentName | char | 40 | NOT NULL | なし |
| DepartmentHeadID | integer | – | NULL | 外部キー |

NOT NULL を指定する場合は、テーブルのどのローにも値がなければなりません。

**参照**

- 「NULL 特別値」『SQL Anywhere サーバー SQL リファレンス』
- 「探索条件」『SQL Anywhere サーバー SQL リファレンス』

## BLOB の考慮事項

BLOB は、未解釈のバイト文字列または文字列で、値としてカラムに格納されます。BLOB の一般的な例としては、画像や音声ファイルがあります。BLOB は大きなものが多く、CHAR、VARCHAR、NCHAR、BINARY、VARBINARY などの文字列やバイナリ文字列のデータ型で格納できます。格納する BLOB の内容や長さに応じて、データ型や長さを選択してください。

**注意**
キャラクターラージオブジェクトは一般に CLOB と呼ばれるのに対し、バイナリラージオブジェクトは BLOB と呼ばれ、両方を組み合わせたものは LOB と呼ばれます。このマニュアルでは、BLOB という略語のみ使用します。

BLOB 値を格納するカラムを作成する場合は、ストレージの特性を制御できます。たとえば、指定したサイズ以下の BLOB をロー (インライン) に格納し、指定したサイズを超える BLOB をロー外のテーブル拡張ページに格納するよう指定できます。また、ロー外に格納した BLOB については、プレフィクスとも呼ばれる先頭の *n* バイトをロー内に複製するよう指定できます。このようなストレージ特性は、CREATE TABLE や ALTER TABLE 文で指定された INLINE や PREFIX 設定によって制御されます。これらの設定で指定した値が、パフォーマンスやディスク記憶領域の要件に予期しない影響が与える場合があります。

INLINE と PREFIX のどちらも指定しない場合、あるいは INLINE USE DEFAULT または PREFIX USE DEFAULT を指定した場合、デフォルト値は次のように適用されます。

- CHAR、NCHAR、LONG VARCHAR、XML などの文字データ型のカラムの場合、INLINE のデフォルト値は 256 で、PREFIX のデフォルト値は 8 です。
- BINARY、LONG BINARY、VARBINARY、BIT、VARBIT、LONG VARBIT、BIT VARYING、UUID などのバイナリデータ型のカラムの場合、INLINE のデフォルト値は 256 で、PREFIX のデフォルト値は 0 です。

デフォルト値では対応できない特定の要件がある場合を除き、INLINE と PREFIX の値は設定しないようにしてください。デフォルト値は、パフォーマンスとディスク領域の要件のバランスを取って選択されています。たとえば、INLINE に大きな値を設定し、すべての BLOB をインラインで格納するようにした場合、ローの処理パフォーマンスは低下することがあります。また、PREFIX の値を大きくしすぎると、BLOB の一部を複製したプレフィクスデータのために、BLOB の格納に必要なディスク領域のサイズが増えることになります。

INLINE や PREFIX の値を設定する場合は、INLINE の長さがカラム長を超えないようにしてください。同様に、PREFIX の長さも INLINE の長さを超えないようにしてください。

圧縮されたカラムのプレフィクスデータは圧縮されずに格納されるため、要求を満たすために必要なすべてのデータがプレフィクス内に格納されている場合は、解凍は必要ありません。

INLINE 句と PREFIX 句のデフォルト値については、「CREATE TABLE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

### BLOB 共有

BLOB のサイズがインラインサイズを超えて、BLOB の格納に 2 ページ以上のデータベースページが必要になった場合、データベースサーバーは同じテーブルの別のローから参照できるよう

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

BLOB を格納します。これは BLOB 共有と呼ばれます。BLOB 共有は内部で処理されます。BLOB 共有はデータベース内で不要に BLOB が複製されないようにするためのものです。

BLOB 共有は、特定のカラムに別のカラムと同じ値を設定する場合にのみ発生します。たとえば UPDATE t column1=column2; のようになります。この例では、column2 に BLOB が含まれている場合に、column1 に BLOB を複製する代わりに、column2 の値を示すように設定します。

BLOB を共有すると、データベースサーバーは BLOB を参照している数を追跡します。データベースサーバーがテーブル内で参照されている BLOB がないと判断すると、BLOB は削除されます。

圧縮されていない 2 つのカラムで BLOB を共有している場合に、一方のカラムを圧縮すると、BLOB は共有でなくなります。

## カラムの圧縮の考慮事項

CHAR、VARCHAR、BINARY カラムを圧縮して、ディスク領域を節約できます。たとえば、BMP や TIFF などの大きい BLOB ファイルを格納するカラムを圧縮できます。圧縮にはデフレート圧縮アルゴリズムが使用されます。これは、COMPRESS 関数で使用される圧縮方式と同じであり、Windows ZIP ファイルで使用されるアルゴリズムでもあります。

圧縮されたカラムは暗号化されたテーブルの内側に格納できます。この場合、データは最初に圧縮されてから暗号化されます。

130 バイト未満の値を含むカラムや、JPG ファイルなどの圧縮形式の値を含むカラムでは、カラムの圧縮を使用しないでください。すでに圧縮されている値を含むカラムを圧縮しようとすると、実際には、カラムに必要な記憶領域のサイズが大きくなる可能性があります。

カラムを圧縮するには、CREATE TABLE 文と ALTER TABLE 文の COMPRESS 句を使用します。

カラムの圧縮による効果を確認するには、sa_column_stats システムプロシージャーを使用します。

**参照**

- 「CREATE TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「ALTER TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「sa_column_stats システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「テーブル暗号化」1405 ページ

## 制約の考慮事項

カラムのデータ型は許可されるデータ値を制限しますが (たとえば数字または日付のみなど)、データ値をさらに制限したい場合もあります。

カラム中のデータ値は検査制約を設定して制限できます。CHECK 制約は、指定された条件をカラムやカラムのセットに課す制約です。WHERE 句に表示される有効な条件を使用して、許可さ

れる値を制限できます。通常、検査制約では BETWEEN または IN 条件を使用します。「カラムに対する検査制約」『SQL Anywhere サーバー SQL の使用法』と「テーブルに対する検査制約」『SQL Anywhere サーバー SQL の使用法』を参照してください。

有効な条件の詳細については、「探索条件」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

テーブルとカラムへの制約の割り当てについては、「データ整合性」『SQL Anywhere サーバー SQL の使用法』を参照してください。

## データベースの作成 (Sybase Central)

**♦ 新規データベースを作成するには、次の手順に従います (Sybase Central の場合)。**

1. **[ツール]** » **[SQL Anywhere 12]** » **[データベースの作成]** をクリックします。
2. **データベース作成ウィザード**の指示に従います。

> **ヒント**
> Sybase Central では、次の方法で**データベース作成ウィザード**にアクセスすることもできます。
> ● サーバーを選択し、**[ファイル]** » **[データベースの作成]** をクリックします。
> ● サーバーを右クリックし、**[データベースの作成]** をクリックします。

### Windows Mobile 用データベースの作成

Windows Mobile 用データベースの作成については、「Windows Mobile データベースサーバーの作成」418 ページを参照してください。

### 参照


- 「SQL Anywhere データベース接続」30 ページ
- 「データベースの作成 (SQL の場合)」118 ページ
- 「データベースの作成 (コマンドラインの場合)」119 ページ


## データベースの作成 (SQL の場合)

**♦ 新規データベースを作成するには、次の手順に従います (SQL の場合)。**

Interactive SQL では、CREATE DATABASE 文を使用してデータベースを作成します。既存のデータベースに接続してから、この文を使用する必要があります。

1. 「sample」という名前のデータベースサーバーを起動します。

```
dbeng12 -n sample
```

2. Interactive SQL を起動します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

3. 既存のデータベースに接続します。データベースが存在しない場合は、ユーティリティデータベース utility_db に接続できます。

4. CREATE DATABASE 文を実行します。

**参照**

- 「ユーティリティデータベース接続」137 ページ
- 「CREATE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「SQL Anywhere データベース接続」30 ページ

**例**

*c:¥temp* ディレクトリにファイル名 *temp.db* のデータベースを作成します。

```
CREATE DATABASE 'c:¥¥temp¥¥temp.db';
```

ディレクトリパスはデータベースサーバーの相対パスになります。この文を実行するのに必要なパーミッションは、-gu dbeng12/dbsrv12 サーバーオプションを使用して設定します。デフォルトの設定は、DBA 権限を必要とします。

円記号 (¥) は SQL のエスケープ文字であるため、場合によって 2 つ付けます。¥x と ¥n の各シーケンスを使用して、16 進数で文字を指定したり、改行文字を指定したりできます。n と x 以外の文字は、前に円記号が付いていても特別な意味はありません。このことが重要になる場合の例を示します。

```
CREATE DATABASE 'c:¥¥temp¥¥¥x41¥x42¥x43xyz.db';
```

最初の ¥¥ シーケンスは円記号を表します。¥x シーケンスは、それぞれ文字 A、B、C を表します。このファイル名は *ABCxyz.db* です。

```
CREATE DATABASE 'c:¥temp¥¥nest.db';
```

¥n シーケンスが改行文字として解釈されないように、円記号を 2 つ連続で使用します。「エスケープシーケンス」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

## データベースの作成 (コマンドラインの場合)

**♦ 新規データベースを作成するには、次の手順に従います (コマンドラインの場合)。**

このユーティリティを使用すると、コマンドラインオプションを含めて別のデータベース設定を指定できます。

- dbinit コマンドを実行します。

  たとえば、ページサイズが 4 KB の *company.db* という名前のデータベースを作成するには、次のコマンドを実行します。

  ```
  dbinit -p 4k company.db
  ```

**参照**

- 「初期化ユーティリティ (dbinit)」958 ページ

## トランザクションログミラーを使用するデータベース

データベース作成時に、トランザクションログミラーも保持するように指定できます。このオプションは、CREATE DATABASE 文、Sybase Central、または dbinit ユーティリティで使用できます。

**♦ トランザクションログミラーを使用するデータベースを作成するには、次の手順に従います (Sybase Central の場合)。**

1. **[ツール]** » **[SQL Anywhere 12]** » **[データベースの作成]** をクリックします。
2. データベース作成ウィザードの指示に従います。

**♦ トランザクションログミラーを使用するデータベースを作成するには、次の手順に従います (SQL の場合)。**

- CREATE DATABASE 文に TRANSACTION LOG 句と MIRROR 句を指定して実行します。次に例を示します。

```
CREATE DATABASE 'c:¥¥mydb.db'
TRANSACTION LOG ON 'mydb.log'
MIRROR 'd:¥¥mydb.mlg';
```

**♦ トランザクションログミラーを使用するデータベースを作成するには、次の手順に従います (コマンドラインの場合)。**

- -m オプションを指定して dbinit ユーティリティを実行します。たとえば、次のコマンド (1 行で入力する) は *company.db* というデータベースを初期化します。トランザクションログは別のデバイスに、そのミラーはさらに別のデバイスに保持されます。

```
dbinit -t d:¥log-dir¥company.log -m
e:¥mirr-dir¥company.mlg c:¥db-dir¥company.db
```

**参照**

- 「CREATE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「初期化ユーティリティ (dbinit)」958 ページ
- 「トランザクションログミラー」128 ページ
- 「SQL Anywhere データベース接続」30 ページ

# 事前定義の DB 領域

SQL Anywhere は、データベースに対して、次に示す事前定義の DB 領域を使用します。

| DB 領域 | 名前 |
|---|---|
| メインデータベースファイル | system |
| テンポラリファイル | temporary または temp |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| DB 領域 | 名前 |
|---|---|
| トランザクションログファイル | translog |
| トランザクションログミラー | translogmirror |

これらの名前を使用して、ユーザー定義の DB 領域を作成することはできません。また、事前定義の DB 領域を削除することはできません。

事前定義の DB 領域と同じ名前が付いたユーザー定義の DB 領域が含まれる、10.0.0 以前のバージョンのデータベースをアップグレードすると、SQL 文でのこれらの DB 領域へのすべての参照は、事前定義の DB 領域ではなく、ユーザー定義の DB 領域を参照していると見なされます。事前定義の DB 領域を参照するように設定するには、ユーザー定義の DB 領域を削除するか、ユーザー定義の DB 領域の名前を変更して事前定義の DB 領域とは別の名前を指定する方法しかありません。

ALTER DBSPACE 文は、事前定義の DB 領域の名前をサポートしているため、これらに対して領域を追加できます。

DB_EXTENDED_PROPERTY 関数も、事前定義の DB 領域の名前を受け入れます。

**参照**

- 「ALTER DBSPACE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「DB_EXTENDED_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』

## 追加の DB 領域の考慮事項

> **注意**
> ほとんどのデータベースでは、データベースファイルは 1 つだけで十分です。しかし、大容量データベースを使用していると、多くの場合、追加データベースファイルが必要になります。また、追加データベースファイルは、別々のファイルにある関連した情報をまとめる場合に便利なツールです。

データベースを初期化すると、データベースにはデータベースファイルが 1 つ含まれます。この最初のデータベースファイルを「メインファイル」または「system」DB 領域と呼びます。デフォルトでは、すべてのデータベースオブジェクトとすべてのデータがこのメインファイルに配置されます。

「DB 領域」は、データ用の領域をさらに作成する追加のデータベースファイルです。1 つのデータベースは 13 個までのファイルに保管されます (メインファイル 1 つと 12 の DB 領域)。各テーブルは、そのインデックスとともに、単一のデータベースファイルに含まれている必要があります。CREATE DBSPACE という SQL 文で、新しいファイルをデータベースに追加できます。

テンポラリテーブルは、temporary DB 領域にのみ作成されます。

ベーステーブルまたはその他のデータベースオブジェクトを作成する DB 領域は複数の方法で指定できます。次のリストで先に示す方法で指定されている場所が、後に示すものよりも優先されます。

1. IN DBSPACE 句 (指定されている場合)
2. default_dbspace オプション (設定されている場合)
3. system DB 領域

DB 領域名にピリオドが含まれていて、引用符で囲まれていない場合、データベースサーバーでエラーが生成されます。

各データベースファイルの最大容量は、$2^{28}$ (約 2 億 6800 万) データベースページです。たとえば、データベースページサイズが 4 KB のデータベースファイルが作成されると、そのファイルのサイズは 1 テラバイト ($2^{28}$*4 KB) まで増やすことができます。しかし実際には、ファイルが作成された物理ファイルシステムで許容される最大ファイルサイズが、最大許容サイズに大きく影響します。

一部の古いファイルシステムではファイルの最大サイズが 2 GB に制限されていますが、Windows が使用している NTFS ファイルシステムのように、多くのファイルシステムでは、データベースファイルを最大サイズまで利用できます。データベースにあるデータの量が最大ファイルサイズを超える場合は、データを複数のデータベースファイルに分割する必要があります。また、関連オブジェクトをまとめる場合など、サイズ制限以外の理由で複数の DB 領域を作成する場合もあります。

サポートされるオペレーティングシステムごとのファイルの最大サイズについては、「SQL Anywhere のサイズと数の制限」811 ページを参照してください。

sa_disk_free システムプロシージャーを使用して、DB 領域に使用可能な領域に関する情報を取得できます。「sa_disk_free_space システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

SYSDBSPACE システムビューには、データベースのすべての DB 領域に関する情報が含まれています。「SYSDBSPACE システムビュー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

### 既存のデータベースの分割

既存のデータベースオブジェクトをいくつかの DB 領域に分割する場合は、データベースをアンロードし、生成済みのスクリプトファイル (デフォルトでは *reload.sql* という名前のファイル) をデータベース再構築用に修正します。*reload.sql* ファイルで、メインファイルに配置しないテーブルごとに、CREATE TABLE 文に IN 句を追加して DB 領域を指定します。

## DB 領域のパーミッション

DB 領域では、CREATE パーミッションのみがサポートされています。CREATE パーミッションによって、ユーザーは指定した DB 領域でデータベースオブジェクトを作成できます。GRANT

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

CREATE ON 文を実行して、DB 領域の CREATE パーミッションを付与できます。「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

DB 領域のパーミッションは、次のように動作します。

- 基本となるデータを使用して新しいオブジェクトを作成するユーザーには、データを配置する DB 領域の CREATE パーミッションが必要です。
- GRANT CREATE ON 文が発行されていても、そのユーザーが新しいデータベースオブジェクトを作成するためにはリソース権限が必要です。
- 特定の DB 領域に配置するため、CREATE パーミッションが必要なオブジェクトの現在のリストには、テーブル、インデックス、テキストインデックス、マテリアライズドビューが含まれています。通常のビューやプロシージャーなどのオブジェクトには、基本となるデータはなく、CREATE パーミッションは必要ありません。
- ユーザーには CREATE パーミッションを直接付与することも、パーミッションが付与されているグループのメンバーシップによってパーミッションを継承させることもできます。
- 特定の DB 領域に対する CREATE パーミッションを PUBLIC に付与することも可能です。この場合、リソース権限を持つすべてのユーザーが DB 領域にオブジェクトを作成できます。
- DB 領域を新規に作成すると、その DB 領域に対する CREATE パーミッションが自動的に PUBLIC に付与されます。
- DB 領域を保護する場合などは、パーミッションを取り消すことができます。内部 DB 領域の system と temporary のパーミッションを管理することで、アクセスを制御することもできます。
- ローカルテンポラリテーブルの作成にはパーミッションは必要なく、DB 領域のパーミッションはローカルテンポラリテーブルの作成には影響しません。ただし、グローバルテンポラリテーブルの作成には、リソース権限と temporary DB 領域に対する CREATE パーミッションが必要です。

**参照**

- 「CREATE DBSPACE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「DB_EXTENDED_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』
- 「CREATE TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「UNLOAD 文」『SQL Anywhere サーバー SQL リファレンス』

## DB 領域の作成

新しいデータベースファイル (DB 領域) は、Sybase Central から、または CREATE DBSPACE 文を使用して作成します。新しい DB 領域のためのデータベースファイルは、メインファイルが存在するディスクドライブ、または別のディスクドライブに配置できます。DB 領域を作成するには DBA 権限が必要です。

各データベースについて、メイン DB 領域に加えて最大 12 の DB 領域を作成できます。新しく作成された DB 領域は空です。新しいテーブルまたはインデックスを作成するときは、CREATE

文に IN 句を指定して特定の DB 領域に配置することもできれば、事前に default_dbspace オプションを設定しておくこともできます。IN 句を指定しなかった場合や default_dbspace オプションの設定を変更しなかった場合、テーブルは system DB 領域に作成されます。

各テーブルは、そのテーブルが作成された DB 領域にすべて格納されます。デフォルトでは、テーブルと同じ DB 領域にインデックスが配置されますが、CREATE 文の一部として IN 句を指定して別の DB 領域に配置することもできます。

**参照**

- 「default_dbspace オプション」617 ページ
- 「CREATE DBSPACE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「CREATE TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「CREATE INDEX 文」『SQL Anywhere サーバー SQL リファレンス』

## DB 領域の作成

**♦ DB 領域を作成するには、次の手順に従います (Sybase Central の場合)。**

1. **SQL Anywhere 12** プラグインを使用して、データベースに接続します。
2. データベースの **[DB 領域]** フォルダーを開きます。
3. **[ファイル]** » **[新規]** » **[DB 領域]** をクリックします。
4. **DB 領域作成ウィザード**の指示に従います。

   新しい DB 領域が **[DB 領域]** フォルダーに表示されます。

**♦ DB 領域を作成するには、次の手順に従います (SQL の場合)。**

- CREATE DBSPACE 文を実行します。

**参照**

- 「CREATE DBSPACE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「default_dbspace オプション」617 ページ
- 「テーブル」『SQL Anywhere サーバー SQL の使用法』
- 「CREATE INDEX 文」『SQL Anywhere サーバー SQL リファレンス』

**例**

次のコマンドは、メインファイルと同じディレクトリにあるファイル *library.db* 内に、MyLibrary という名前の新しい DB 領域を作成します。

```
CREATE DBSPACE MyLibrary
AS 'library.db';
```

次のコマンドは、LibraryBooks テーブルを作成し、それを MyLibrary DB 領域に配置します。

```
CREATE TABLE LibraryBooks (
title CHAR(100),
```

```
author CHAR(50),
isbn CHAR(30)
) IN MyLibrary;
```

次のコマンドは、MyLibrary という名前の新しい DB 領域を作成し、この DB 領域をデフォルト DB 領域に設定したうえで、この DB 領域に LibraryBooks テーブルを作成します。

```
CREATE DBSPACE MyLibrary
AS 'e:¥¥dbfiles¥¥library.db';
SET OPTION default_dbspace = 'MyLibrary';
CREATE TABLE LibraryBooks (
  title CHAR(100),
  author CHAR(50),
  isbn CHAR(30),
);
```

## データベースファイル領域の事前割り付け

新しいデータベースファイルを作成するときは、CREATE DATABASE 文の DATABASE SIZE 句を使用するか、dbinit -dbs オプションを指定することによって、データベース領域を事前に割り付けることができます。「CREATE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』と「初期化ユーティリティ (dbinit)」958 ページを参照してください。

データベースを使用していると、必要に応じてデータベースファイルのサイズが自動的に増大します。データベースファイルを頻繁に更新していると、ディスク上のファイルが過度に断片化し、パフォーマンスが低下することがあります。サイズの小さな多数の領域を割り付けるには、サイズの大きい領域を 1 つ割り付けるよりも時間がかかります。変更の頻度が高いデータベースの場合は、Sybase Central または ALTER DBSPACE 文を使用して、DB 領域やトランザクションログに対し、ディスク領域を事前に割り付けることができます。「ALTER DBSPACE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

**♦ 領域を事前に割り付けるには、次の手順に従います (Sybase Central の場合)。**

> **ヒント**
> ディスク領域を事前に割り付けてからディスク断片化解除ユーティリティを実行すると、ディスクドライブのさまざまな箇所にデータベースファイルが断片化されるのを、確実に防ぐことができます。データベースファイルの断片化が進むと、パフォーマンスが低下します。

1. DBA 権限のあるユーザーとしてデータベースに接続します。
2. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。
3. **[DB 領域]** フォルダーを開きます。
4. DB 領域を右クリックし、**[領域の事前割り付け]** をクリックします。
5. DB 領域に追加する領域のサイズを入力します。領域は、ページ、バイト、キロバイト (KB)、メガバイト (MB)、ギガバイト (GB)、またはテラバイト (TB) 単位で追加できます。
6. **[OK]** をクリックします。

**♦ 領域を事前に割り付けるには、次の手順に従います (SQL の場合)。**

> **ヒント**
> ディスク領域を事前に割り付けてからディスク断片化解除ユーティリティを実行すると、ディスクドライブのさまざまな箇所にデータベースファイルが断片化されるのを、確実に防ぐことができます。データベースファイルの断片化が進むと、パフォーマンスが低下します。

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. ALTER DBSPACE 文を実行します。

**参照**

- 「DB 領域の作成」 123 ページ
- 「ALTER DBSPACE 文」『SQL Anywhere サーバー SQL リファレンス』

**例**

system DB 領域のサイズを 200 ページ増やします。

```
ALTER DBSPACE system
ADD 200;
```

system DB 領域のサイズを 400 メガバイト増やします。

```
ALTER DBSPACE system
ADD 400 MB;
```

## DB 領域の削除

**♦ DB 領域を削除するには、次の手順に従います (Sybase Central の場合)。**

DB 領域を削除する前に、その DB 領域を使用するテーブルとインデックスをすべて削除する必要があります。

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. **SQL Anywhere 12** プラグインを使用して、データベースに接続します。
3. **[DB 領域]** フォルダーを開きます。
4. DB 領域を右クリックし、**[削除]** をクリックします。

**♦ DB 領域を削除するには、次の手順に従います (SQL の場合)。**

DB 領域を削除する前に、その DB 領域を使用するテーブルとインデックスをすべて削除する必要があります。

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. DROP DBSPACE 文を実行します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照

- 「DROP TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「DROP DBSPACE 文」『SQL Anywhere サーバー SQL リファレンス』

## トランザクションログ

「トランザクションログ」は、データベースファイルとは別のファイルです。トランザクションログは、データベースに対して行われたすべての変更を格納します。挿入、更新、削除、コミット、ロールバック、データベーススキーマの変更が、すべて記録されます。トランザクションログは、「転送ログ」または「REDO ログ」とも呼ばれます。

トランザクションログは、バックアップとリカバリの重要なコンポーネントであり、Mobile Link を使用したデータ同期や、SQL Remote を使用したデータレプリケーションにも不可欠です。

デフォルトでは、すべてのデータベースがトランザクションログを使用します。トランザクションログの使用はオプションですが、特別の理由がないかぎり、トランザクションログの使用をおすすめします。データベースの実行時にトランザクションログを使用すると、障害からの保護が強化され、パフォーマンスが向上し、データをレプリケートできるようになります。

データベースファイルとトランザクションログを、コンピューターの別々のディスクに保存することをおすすめします。DB 領域とトランザクションログが同じディスク上にある状態でディスク障害が発生すると、すべて失われる可能性があります。しかし、データベースとトランザクションログが別々のディスクに保存されている場合は、ディスク障害が発生した場合でも、すべてまたはほとんどのデータをリカバリできます。これは、フルデータベースや、データベースをリカバリするベースとなるトランザクションログが存在するからです。「メディア障害」1106 ページを参照してください。

データベースファイルまたはトランザクションログファイルのタイムスタンプは、ファイルが増大するか、またはファイルが閉じられた場合にのみ更新されます。データベース操作によってトランザクションログファイルが増大するが、データベースファイルが増大しない場合は、トランザクションログファイルのタイムスタンプがデータベースファイルのタイムスタンプよりも最近のものになります。データベースが停止すると、トランザクションログファイルとデータベースファイルのタイムスタンプが更新されます。

**警告**
データベースファイルとトランザクションログファイルは、データベースサーバーと同じ物理コンピューターに保存してください。または SAN や iSCSI 設定でアクセスできるようにしてください。リモートネットワークディレクトリにデータベースファイルやトランザクションログファイルを配置すると、パフォーマンスが低下したり、データが破壊されたり、サーバーが不安定になったりする可能性があります。http://www.sybase.com/detail?id=1034790 を参照してください。

### 変更がディスクに書き込まれるとき

データベースファイルと同様に、トランザクションログは「ページ」、つまり固定サイズのメモリ領域に保持されます。トランザクションログに変更が記録されるとき、その内容はメモリ内の

ページに書き込まれます。変更は、次の状況のいずれかが発生した段階で、強制的にディスクに書き込まれます。

- ページが満杯になったとき
- COMMIT が実行されたとき

完了したトランザクションは確実にディスクに格納され、また操作のたびにディスクに書き込む必要がないため、パフォーマンスも向上します。

設定オプションを使用すると、詳しい知識があるユーザーはトランザクションログの動作を細かくチューニングできます。 「cooperative_commits オプション」611 ページと「delayed_commits オプション」620 ページを参照してください。

**参照**


- 「トランザクションログサイズの考慮事項」130 ページ
- 「-m dbeng12/dbsrv12 サーバーオプション」255 ページ
- 「-m dbeng12/dbsrv12 データベースオプション」311 ページ
- 「sa_disk_free_space システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「delete_old_logs オプション [SQL Remote]」621 ページ


## トランザクションログミラー

「トランザクションログミラー」はトランザクションログの完全なコピーであり、トランザクションログと同時に管理されます。データベースにトランザクションログミラーがある場合、データベースに対する変更は、トランザクションログとトランザクションログミラーの両方に書き込まれます。デフォルトでは、データベースは、トランザクションログミラーを持ちません。

トランザクションログミラーは、重要なデータを二重に保護します。トランザクションログでメディア障害が発生した場合に、完全なデータリカバリが可能になります。また、トランザクションログミラーがあると、データベースの開始時に、データベースサーバーによるトランザクションログの自動検証が可能になります。

大容量または重要なアプリケーションを実行する場合は、トランザクションログミラーを使用することをおすすめします。たとえば、SQL Remote 設定における統合データベースでは、レプリケーションはトランザクションログに依存します。トランザクションログが損傷した場合、データのレプリケーションは失敗します。

トランザクションログミラーを使用している場合、ログのいずれかに書き込もうとしてエラーが発生すると (ディスクが満杯の場合など)、データベースサーバーが停止します。トランザクションログミラーの目的は、いずれかのログデバイスでメディア障害が発生したときに完全にリカバリできるようにすることです。1 つのトランザクションログを使用してサーバーを実行し続けると、この目的は達成できません。

データベースサーバーの起動時に -fc オプションを指定して、データベースサーバーでファイルシステムがいっぱいになった場合のコールバック関数を実装できます。「-fc dbeng12/dbsrv12 サーバーオプション」229 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### トランザクションログミラーの保存先

トランザクションログミラーを使用すると、各データベースログへの書き込みを 2 回ずつ行う必要があるため、パフォーマンスが低下します。低下の程度は、データベース内のデータ転送の形態と量、データベースとログの物理的な設定の方法によって異なります。

トランザクションログミラーは、トランザクションログとは別のデバイスに保管してください。これによってパフォーマンスが向上し、また一方のデバイスが故障しても、ログのもう一方のコピーにリカバリのためのデータが残ります。

### トランザクションログミラーに代わる方法

トランザクションログミラーに代わる方法として、次の構成を使用できます。

- データベースのミラーリング。「データベースミラーリング」1137 ページを参照してください。

- ハードウェアミラーリングを行うディスクコントローラ。通常、ハードウェアミラーリングはオペレーティングシステムレベルのソフトウェアミラーリングよりもコストがかかりますが、パフォーマンスは高くなります。

- Microsoft Windows に用意されているオペレーティングシステムレベルのソフトウェアミラーリング。

ライブバックアップを使用すると、トランザクションログミラーに似た方法でデータを保護することができます。「ライブバックアップとトランザクションログミラーの違い」1076 ページを参照してください。

トランザクションログミラーを使用するデータベースを作成する方法については、「初期化ユーティリティ (dbinit)」958 ページを参照してください。

トランザクションログミラーを使用するように既存のデータベースを変更する方法については、「トランザクションログユーティリティ (dblog)」1041 ページを参照してください。

## トランザクションログの場所の変更

♦ **トランザクションログの場所を変更するには、次の手順に従います (Sybase Central の場合)。**

トランザクションログの場所を変更する場合、データベースは停止している必要があります。

トランザクションログの格納場所を選択する方法の詳細については、「トランザクションログ」127 ページを参照してください。

1. **SQL Anywhere 12** プラグインを使用して、データベースに接続します。
2. **[ツール]** » **[SQL Anywhere 12]** » **[ログファイル設定の変更]** をクリックします。
3. **ログファイル設定の変更ウィザード**の指示に従います。

**♦ 既存のデータベースのトランザクションログミラーの場所を変更するには、次の手順に従います (コマンドラインの場合)。**

トランザクションログの場所を変更する場合、データベースは停止している必要があります。

1. データベースサーバーが起動していないことを確認します。
2. 次のコマンドを実行します。

```
dblog -t new-transaction-log-file database-file
```

**参照**

- 「トランザクションログユーティリティ (dblog)」1041 ページ

## 既存のデータベースのトランザクションログミラー

データベースが実行中ではないときはいつでもトランザクションログユーティリティを使用して、既存のデータベースのトランザクションログミラーを管理できます。

dblog ユーティリティと Sybase Central を使用して、データベースでトランザクションログミラーを使用しないようにすることもできます。

**♦ 既存のデータベースでトランザクションログミラーを開始するには、次の手順に従います (Sybase Central の場合)。**

1. **SQL Anywhere 12** プラグインを使用して、データベースに接続します。
2. **[ツール]** » **[SQL Anywhere 12]** » **[ログファイル設定の変更]** をクリックします。
3. **ログファイル設定の変更ウィザード**の指示に従います。

**♦ 既存のデータベースに対してトランザクションログミラーを開始するには、次の手順に従います (コマンドラインの場合)。**

1. データベースサーバーが起動していないことを確認します。
2. 次のコマンドを実行します。

```
dblog -m mirror-file database-file
```

**参照**

- 「トランザクションログユーティリティ (dblog)」1041 ページ

## トランザクションログサイズの考慮事項

トランザクションログのサイズは、リカバリの所要時間に影響します。すべてのテーブルに簡単なプライマリキーを設定することによって、トランザクションログファイルの拡大を制御できま

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

す。プライマリキーまたは NULL 入力不可のユニークインデックスがないテーブルで更新または削除を実行すると、対象ローの内容がすべてトランザクションログに保存されます。プライマリキーが定義されている場合は、データベースサーバーはそのカラム値を保存するだけでローをユニークに識別できます。テーブルのカラム数が多い場合、またはテーブルのカラムに長いデータを含む場合は、プライマリキーが定義されていないと、トランザクションログのページがすぐに満杯になります。このように、データの余分な書き込みによって、ディスク領域を多く必要とするだけではなく、パフォーマンスが低下します。

プライマリキーがない場合、サーバーはテーブル上で UNIQUE NOT NULL インデックス (または UNIQUE 制約) を探します。NULL 値を許容する UNIQUE インデックスでは十分に識別できないためです。

**参照**


- 「-m dbeng12/dbsrv12 サーバーオプション」255 ページ
- 「-m dbeng12/dbsrv12 データベースオプション」311 ページ
- 「sa_disk_free_space システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「delete_old_logs オプション [SQL Remote]」621 ページ


## 接続の未処理のトランザクション

トランザクションログの名前変更または削除を行うバックアップを実行すると、完了していないトランザクションは、新しいトランザクションログに持ち越されます。

システムプロシージャーを使用して、未処理のトランザクションがあるユーザーを判別できます。接続がそれほど多くない場合は、SQL Anywhere コンソールユーティリティを使用して未処理のトランザクションがある接続を確認することもできます。必要に応じて、DROP CONNECTION 文を使用してユーザーを切断できます。

**♦ 未処理のトランザクションがある接続を判断するには、次の手順に従います (SQL の場合)。**

1. Interactive SQL からデータベースに接続します。
2. sa_conn_info システムプロシージャーを実行します。

   ```
   CALL sa_conn_info;
   ```

3. **UncommitOps** カラムを検査して、未処理のトランザクションがある接続を確認します。

   「sa_conn_info システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

**♦ 未処理のトランザクションがある接続を判断するには、次の手順に従います (SQL Anywhere コンソールユーティリティの場合)。**

1. SQL Anywhere コンソールユーティリティからデータベースに接続します。

   たとえば、次のコマンドでは、ユーザー ID DBA とパスワード sql を使用してデフォルトデータベースに接続します。

```
dbconsole -c "UID=DBA;PWD=sql"
```

「SQL Anywhere コンソールユーティリティ (dbconsole)」1021 ページを参照してください。

2. 各接続をダブルクリックし、**[コミットされていないオペレーション]** のエントリを調べて、コミットされていない操作のあるユーザーを確認します。

**参照**

- 「DROP CONNECTION 文」『SQL Anywhere サーバー SQL リファレンス』

## チェックポイント

「チェックポイント」とは、ダーティページがすべてディスクに書き込まれる時点のことで、ディスク上にあるデータベースの既知の一貫性のある状態を示します。チェックポイントの後で、チェックポイントログの内容が削除されます。空のチェックポイントログページは、現在のセッション中はチェックポイントログに存在し、新しいチェックポイントログデータに再利用できます。チェックポイントログのサイズが大きくなると、データベースファイルも大きくなります。

チェックポイント時には、データベースの全データがディスクのデータベースファイルに格納されます。データベースファイルの情報は、トランザクションログの情報と一致します。リカバリ時には、データベースはまず最新のチェックポイントでリカバリされ、次にチェックポイント以降の変更内容が適用されます。

各セッションの最後では、チェックポイントログの使用履歴がデータベースに格納され、次のセッションのチェックポイントログに適切なサイズを決定するために使用されます。

データベースサーバーは、チェックポイントを開始し、チェックポイントの発生中にその他の操作を実行できます。ただし、チェックポイントがすでに進行中だった場合、新しいチェックポイントを開始する文は、現在のチェックポイントの完了を待ってからでないと実行できません。

**次の文は、暗黙的または明示的なチェックポイントを開始します。**

- ALTER INDEX REBUILD 文。「ALTER INDEX 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

- ALTER TABLE 文 – ALTER TABLE 文が NULL 入力可のカラムまたは参照整合性制約を追加せず、テーブルのページを物理的に変更しない場合。「ALTER TABLE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

- BACKUP 文。「BACKUP 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

- CHECKPOINT 文。「CHECKPOINT 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

- COMMIT 文 – データベースがトランザクションログを作成していないとき。「COMMIT 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

●CREATE DBSPACE 文。「CREATE DBSPACE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

●CREATE INDEX 文 – 関数にインデックスを作成するために使用されるとき (暗黙の計算カラム)。「CREATE INDEX 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

●DROP DBSPACE 文。「DROP DBSPACE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

●DROP TABLE 文と DROP MATERIALIZED VIEW 文 – テーブルまたはビューに少なくとも 1 つのローが含まれている場合のみ。「DROP TABLE 文」『SQL Anywhere サーバー SQL リファレンス』と「DROP MATERIALIZED VIEW 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

●LOAD TABLE 文。「LOAD TABLE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

●REFRESH MATERIALIZED VIEW 文。「REFRESH MATERIALIZED VIEW 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

●REORGANIZE TABLE 文。「REORGANIZE TABLE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

**参照**


- 「バックアップとリカバリの制限」1079 ページ
- 「バックアップコンポーネント」1113 ページ
- 「データベースサーバーがチェックポイントのタイミングを決定する方法」1114 ページ
- 「チェックポイントログ」133 ページ

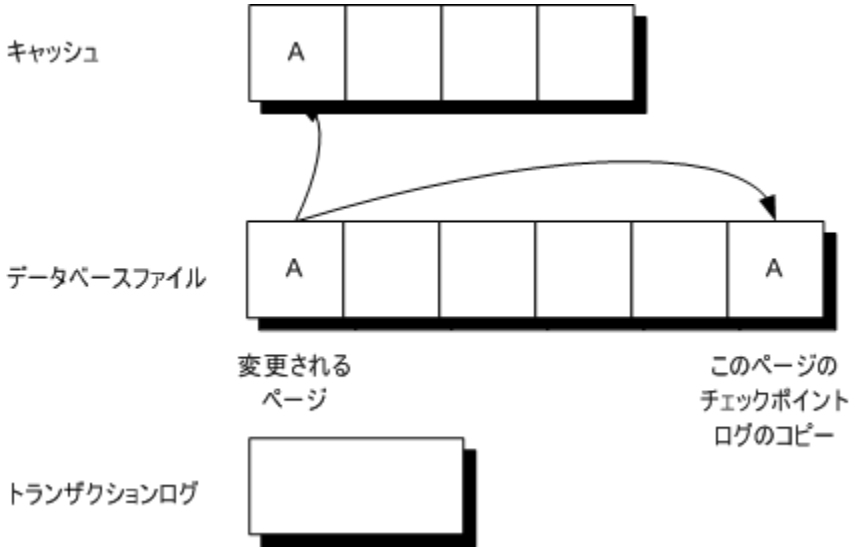

## チェックポイントログ

データベースファイルはページで構成されています。ページとは、ハードディスク内の決められたサイズの領域です。「チェックポイントログ」は、データベースファイルの末尾にあり、system DB 領域に保存されます。セッション中とセッションの最後には必要に応じてチェックポイントログにページが追加され、チェックポイントの使用履歴がデータベースに格納されます。この履歴は、以降のセッションのチェックポイントログに適切なサイズを決定するために使用されます。

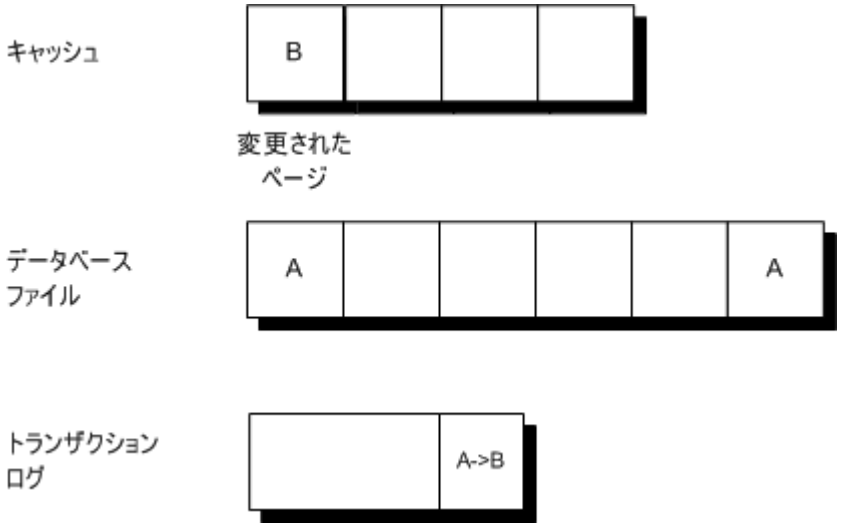
データベースサーバーでは、ページが更新される (「ダーティ」になる) 前に次の処理が実行されます。

●ページがメモリに読み込まれ、データベースキャッシュ内に格納されます。

●元のページのコピーが作成されます。これらのコピーされたページを「チェックポイントログ」と呼びます。

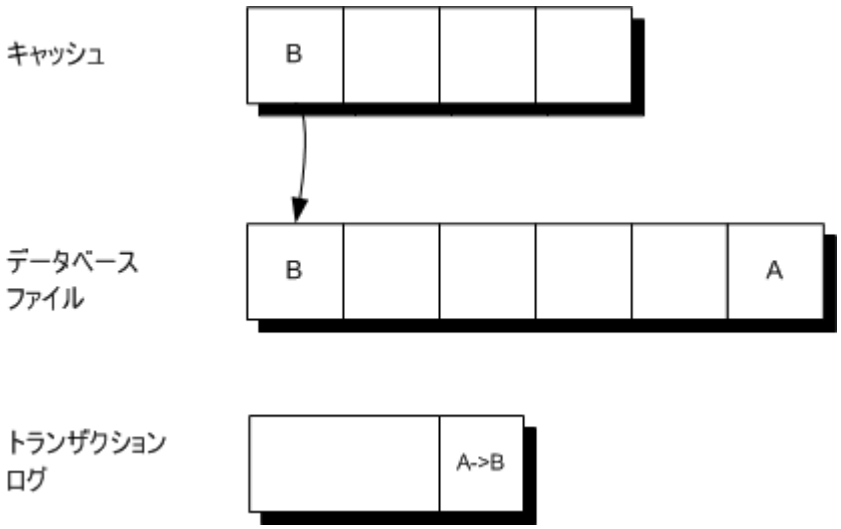



ページに加えた変更は、キャッシュ内のコピーに適用されます。パフォーマンス上の理由から、ページはすぐにはディスクのデータベースファイルに書き込まれません。




キャッシュが満杯になると、変更されたページはディスクに書き込まれます。チェックポイントログのコピーは変更されません。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1




**参照**

- 「バックアップとリカバリの制限」1079 ページ
- 「バックアップコンポーネント」1113 ページ
- 「データベースサーバーがチェックポイントのタイミングを決定する方法」1114 ページ
- 「チェックポイント」132 ページ

# ユーティリティデータベース

「ユーティリティデータベース」は、物理的な実体を持たない幻データベースです。この機能によって、CREATE DATABASE などのデータベースファイル管理文を、既存の物理データベースに接続しなくても実行できます。ユーティリティデータベースにはデータベースファイルがないため、データを入れることができません。

ユーティリティデータベースの名前は **utility_db** です。この名前を持つデータベースを作成または起動しようとすると、操作は失敗します。

ユーティリティデータベースに接続してから次の文を実行すると、*new.db* という名前のデータベースがディレクトリ *c:¥temp* に作成されます。

```
CREATE DATABASE 'c:¥¥temp¥¥new.db';
```

「CREATE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

ユーティリティデータベースを使用して、接続プロパティとサーバープロパティの値を取り出すこともできます。

たとえば、ユーティリティデータベースで次の文を実行すると、デフォルトの照合順が返され、作成するデータベースに使用できます。

```
SELECT PROPERTY( 'DefaultCollation' );
```

接続プロパティとデータベースサーバープロパティの詳細については、次の各項を参照してください。


- 「接続プロパティ値のアクセス」713 ページ
- 「データベースサーバープロパティ値のアクセス」753 ページ


ミラーリングシステムでは、データベースサーバーが実行中であったとしてもデータベースに接続できないことがあります。そのような場合、ユーティリティデータベースに接続してサーバーをシャットダウンしたり、データベースを強制的にプライマリサーバーにしたりすることができます。「ミラーリングシステムのデータベースサーバーの停止」1168 ページと「データベースサーバーの強制プライマリサーバー化」1166 ページを参照してください。

### ユーティリティデータベースに使用できる文

ユーティリティデータベースに接続している場合は、次の文のみを実行できます。

- ALTER DATABASE *dbfile* ALTER TRANSACTION LOG。「ALTER DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。
- CREATE DATABASE 文。「CREATE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。
- CREATE DECRYPTED DATABASE 文。「CREATE DECRYPTED DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。
- CREATE DECRYPTED FILE 文。「CREATE DECRYPTED FILE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。
- CREATE ENCRYPTED DATABASE 文。「CREATE ENCRYPTED DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。
- CREATE ENCRYPTED FILE 文。「CREATE ENCRYPTED FILE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。
- DROP DATABASE 文。「DROP DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。
- CREATE USER DBA IDENTIFIED BY *new-password*。「CREATE USER 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。
- RESTORE DATABASE 文。「RESTORE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。
- REVOKE CONNECT FROM DBA。「REVOKE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。
- SET TEMPORARY OPTION progress_messages = [OFF | RAW | FORMATTED]。「SET OPTION 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。
- FROM 句または WHERE 句なしの SELECT 文。「SELECT 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。
- START DATABASE 文。「START DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。
- STOP DATABASE 文。「STOP DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。
- STOP SERVER 文。「STOP SERVER 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

### 参照


- 「ファイル管理文の実行に必要なパーミッション」561 ページ


 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## ユーティリティデータベース接続

サーバーへの接続時に utility_db をデータベース名として指定すると、データベースサーバー上のユーティリティデータベースを起動できます。-su サーバーオプションを使用すると、DBA ユーザーのユーティリティデータベースパスワードを設定したり、ユーティリティデータベースへの接続を無効にしたりできます。-su オプションを指定しないでユーティリティデータベースを起動すると、パーソナルサーバーの場合とネットワークサーバーの場合では、ユーザー ID やパスワードの要件が変わります。

- パーソナルデータベースサーバーの場合、-su オプションが指定されないと、ユーティリティデータベースへの接続に関するセキュリティ上の制限はありません。パーソナルサーバーの場合、ユーザー ID として DBA を指定します。また、パスワードも指定する必要がありますが、どのようなパスワードでもかまいません。パーソナルデータベースサーバーに接続できるユーザーであればファイルシステムに直接アクセスできると想定されているので、パスワードによるユーザーの選別は行われません。

  ユーティリティデータベースのパスワードをプレーンテキストで入力することを回避するには、-su オプションを使用してパスワードを含むファイルを作成し、dbfhide ユーティリティを使用してファイルの内容を難読化します。たとえば、ユーティリティデータベースのパスワードを含むファイルの名前が *util_db_pwd.cfg* であるとします。dbfhide を使用してこのファイルを難読化し、*util_db_pwd_hide.cfg* という名前に変更します。

  ```
  dbfhide util_db_pwd.cfg util_db_pwd_hide.cfg
  ```

  その後、*util_db_pwd_hide.cfg* ファイルを使用して、ユーティリティデータベースのパスワードを指定できます。

  ```
  dbsrv12 -su @util_db_pwd_hide.cfg -n my_server c:¥mydb.db
  ```

  「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。

- ネットワークサーバーの場合、-su オプションが指定されないと、ユーザー ID として DBA を指定する必要があります。また、データベースサーバーの実行ファイルと同じディレクトリにある *util_db.ini* ファイルに格納されているパスワードも指定します。このディレクトリはサーバー上にあるため、ファイルへのアクセスを制御でき、パスワードを使用できるユーザーも制御することができます。パスワードでは大文字と小文字が区別されます。

  **注意**
  *util_db.ini* ファイルの使用は推奨されません。-su サーバーオプションを使用して、ユーティリティデータベースの DBA ユーザーのパスワードを指定してください。「-su dbeng12/dbsrv12 サーバーオプション」276 ページを参照してください。

  ネットワークデータベースサーバー用のユーティリティデータベースにパスワードを指定することは、データベースサーバーが実行中の場合には便利ですが、データベースに接続できません。たとえば、ミラーリングシステムでは、ユーティリティデータベースに接続してデータベースサーバーをシャットダウンしたり、必要に応じてミラーサーバーを強制的にプライマリサーバーにしたりできるようになります。「ミラーリングシステムのデータベースサーバーの停止」1168 ページと「データベースサーバーの強制プライマリサーバー化」1166 ページを参照してください。

**♦ パーソナルサーバー上のユーティリティデータベースに接続するには、次の手順に従います (Interactive SQL の場合)。**

> **注意**
> ユーティリティデータベースに接続するときに REVOKE CONNECT FROM DBA を実行すると、以降のユーティリティデータベースへの接続が無効になります。これは、REVOKE CONNECT を実行する前に確立していた接続を使用するか、データベースサーバーを再起動しないかぎり、以降はユーティリティデータベースに接続できないことを意味します。「REVOKE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

1. 次のコマンドを使用してデータベースサーバーを起動します。

   ```
   dbeng12 -n TestEng
   ```

   セキュリティを高めるため、-su オプションを使用して、ユーティリティデータベースパスワードを指定します。

2. Interactive SQL を起動します。

3. **[接続]** ウィンドウで、次の情報を指定します。
   a. **[ユーザー ID]** フィールドに **DBA** と入力します。
   b. **[パスワード]** フィールドに、空白でないパスワードを入力します。パスワード自体は確認されませんが、フィールドを空白にすることはできません。
   c. **[アクション]** ドロップダウンリストから、**[このコンピューターで稼働しているデータベースに接続]** をクリックします。

4. **[データベース名]** フィールドに **utility_db** と入力します。

5. **[サーバー名]** フィールドに **TestEng** と入力します。

6. **[接続]** をクリックします。

   Interactive SQL は、TestEng というパーソナルサーバー上のユーティリティデータベースに接続します。

**♦ ネットワークサーバー上のユーティリティデータベースに接続するには、次の手順に従います (Interactive SQL の場合)。**

> **注意**
> ユーティリティデータベースに接続するときに REVOKE CONNECT FROM DBA を実行すると、以降のユーティリティデータベースへの接続が無効になります。これは、REVOKE CONNECT を実行する前に確立していた接続を使用するか、データベースサーバーを再起動しないかぎり、以降はユーティリティデータベースに接続できないことを意味します。「REVOKE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

1. 次のコマンドを使用してデータベースサーバーを起動します。

   ```
   dbsrv12 -n TestEng -su 9Bx231K
   ```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

2. Interactive SQL を起動します。
3. **[接続]** ウィンドウで、**[ユーザー ID]** に **DBA** と入力し、-su オプションで指定したパスワードを入力します。この例では、パスワードは **9Bx231K** です。
4. **[アクション]** ドロップダウンリストで、**[別のコンピューターで稼働しているデータベースに接続]** をクリックします。
5. **[データベース名]** フィールドに **utility_db** と入力します。
6. **[サーバー名]** フィールドに **TestEng** と入力します。
7. **[接続]** をクリックします。

   Interactive SQL は、TestEng というネットワークサーバー上のユーティリティデータベースに接続します。

### ネットワークデータベースサーバーでの *util_db.ini* の使用 (旧式)

> **注意**
> *util_db.ini* ファイルの使用は推奨されません。-su サーバーオプションを使用して、ユーティリティデータベースの DBA ユーザーのパスワードを指定してください。「-su dbeng12/dbsrv12 サーバーオプション」276 ページを参照してください。

*util_db.ini* の使用は、データベースサーバーを実行するコンピューターの物理的なセキュリティに依存します。これは、テキストエディターを使用して *util_db.ini* ファイルを簡単に読み取れるからです。

ネットワークサーバーの場合、デフォルトでは、-su オプションまたは *util_db.ini* を使用しないでユーティリティデータベースに接続することはできません。*util_db.ini* を使用する場合、このファイルにパスワードが格納されています。このファイルは、データベースサーバーの実行プログラムと同じディレクトリに配置され、次のテキストを含んでいます。

```
[UTILITY_DB]
PWD=password
```

不用意な直接アクセスから *util_db.ini* ファイルの内容を保護するには、ファイル難読化ユーティリティ (dbfhide) を使って、ファイルに単純暗号化を追加します。オペレーティングシステムの機能を使用して、サーバーのファイルシステムへのアクセスを制限することもできます。

.INI ファイルの難読化の詳細については、「.ini ファイルの内容の非表示」465 ページを参照してください。

### 参照


- 「SQL Anywhere データベース接続」5 ページ
- 「-su dbeng12/dbsrv12 サーバーオプション」276 ページ

# データベースの消去

データベースを消去すると、データベースへの変更を記録したトランザクションログを含むすべてのテーブルとデータがディスクから削除されます。データベースファイルはすべて、誤ってファイルを変更したり削除したりするのを防ぐために読み込み専用となっています。

dberase ユーティリティを使用して、コマンドラインからデータベースを消去する方法もあります。ただし、dberase ユーティリティは DB 領域を消去しません。DB 領域を消去したい場合、DROP DATABASE 文を使用するか、または Sybase Central の**データベース消去ウィザード**を使用します。

dberase ユーティリティ、**データベース消去ウィザード**、DROP DATABASE 文を使用する場合、消去するデータベースが実行中であってはなりません。別のデータベースを削除するには、データベースに接続されている必要があります。

ユーティリティデータベースへの接続方法については、「ユーティリティデータベース接続」137 ページを参照してください。

Windows Mobile のデータベースは手動で消去する必要があります。「Windows Mobile データベースの消去」426 ページを参照してください。

**♦ データベースを消去するには、次の手順に従います (Sybase Central の場合)。**

データベースを消去するには、デフォルトで DBA 権限が必要です。データベースサーバーの -gu オプションを使用すると、必要なパーミッションを変更できます。「-gu dbeng12/dbsrv12 サーバーオプション」247 ページを参照してください。

1. DBA 権限を持つユーザーとして消去対象でないデータベースに接続します。たとえば、ユーティリティデータベースに接続します。
2. **[ツール]** » **[SQL Anywhere 12]** » **[データベースの消去]** をクリックします。
3. ウィザードの指示に従います。

**ヒント**
Sybase Central では、次の方法で**データベース消去ウィザード**にアクセスすることもできます。

●データベースサーバーを選択し、**[ファイル]** » **[データベースの消去]** をクリックします。

●サーバーを右クリックし、**[データベースの消去]** をクリックします。

**♦ データベースを消去するには、次の手順に従います (SQL の場合)。**

データベースを消去するには、デフォルトで DBA 権限が必要です。データベースサーバーの -gu オプションを使用すると、必要なパーミッションを変更できます。「-gu dbeng12/dbsrv12 サーバーオプション」247 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

1. DBA 権限を持つユーザーとして消去対象でないデータベースに接続します。たとえば、ユーティリティデータベースに接続します。
2. DROP DATABASE 文を実行します。

   たとえば、次の DROP DATABASE 文は、temp というデータベースを消去します。

   ```
   DROP DATABASE 'c:¥¥temp¥¥temp.db';
   ```

   「DROP DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

**♦ データベースを消去するには、次の手順に従います (コマンドラインの場合)。**

データベースを消去するには、デフォルトで DBA 権限が必要です。データベースサーバーの -gu オプションを使用すると、必要なパーミッションを変更できます。

dberase ユーティリティを使用して、コマンドラインからデータベースを消去する方法もあります。ただし、dberase ユーティリティは DB 領域を消去しません。DB 領域を消去したい場合、DROP DATABASE 文を使用するか、または Sybase Central の**データベース消去ウィザード**を使用します。

- dberase ユーティリティを実行します。

  たとえば、次のコマンドは、temp データベースを削除します。

  ```
  dberase c:¥temp¥temp.db
  ```

**参照**


- 「-gu dbeng12/dbsrv12 サーバーオプション」247 ページ
- 「消去ユーティリティ (dberase)」951 ページ


# SQL Anywhere データベースサーバーの使用

**SQL Anywhere データベースサーバー：**

SQL Anywhere は、「パーソナルサーバー」と「ネットワークサーバー」の 2 つのデータベースサーバーを提供しています。パーソナルデータベースサーバーで作成されたデータベースは、ネットワークデータベースサーバーで使用でき、逆についても同様です。パーソナルサーバーとネットワークサーバーでは、同じ要求処理エンジンが使用され、どちらも同一の SQL 言語と多くの同じデータベース機能がサポートされています。ただし、この 2 つのサーバーには 2、3 の違いがあります。

- **パーソナルデータベースサーバー**　この実行プログラムは、単一ユーザーによる単一コンピューターでの使用を目的として、たとえば、組み込みデータベースサーバーとして提供されており、ネットワーク経由のクライアント／サーバー通信はサポートしていません。

  Windows Mobile 以外の Windows オペレーティングシステムでは、パーソナルサーバーの実行プログラムの名前は *dbeng12.exe* です。UNIX オペレーティングシステムでは、*dbeng12* です。パーソナルサーバーは、Windows Mobile ではサポートされていません。

- **ネットワークデータベースサーバー**　この実行プログラムは、ネットワーク経由のクライアント／サーバー通信をサポートし、複数ユーザーでの使用を目的としています。

  Windows Mobile を含む Windows オペレーティングシステムでは、ネットワークサーバーの実行プログラムの名前は *dbsrv12.exe* です。Linux と UNIX オペレーティングシステムでは、*dbsrv12* です。

### 必要なネットワークソフトウェア

SQL Anywhere ネットワークサーバーを実行している場合、適切なネットワークソフトウェアがインストールされて実行されている必要があります。

SQL Anywhere ネットワークサーバーは、Windows、Linux、UNIX、Mac OS X で使用できます。

SQL Anywhere では TCP/IP ネットワーク プロトコルもサポートされています。

### 参照


- 「SQL Anywhere データベースサーバーの構文」189 ページ
- 「データベースサーバーの違い」142 ページ


## データベースサーバーの違い

次の表は、一般的なデータベースサーバーの機能と、パーソナルデータベースサーバーとネットワークデータベースサーバーとの間でどのような違いがあるのかを一覧で示しています。

| 機能 | ネットワークデータベースサーバー (dbsrv12) | パーソナルデータベースサーバー (dbeng12) |
|---|---|---|
| チェックポイント | CHECKPOINT 文を使用するには DBA 権限が必要です。 | CHECKPOINT 文を使用するために必要なパーミッションは何もありません。 |
| 通信パケットの圧縮 | サポートされています。 | サポートされていません。 |
| ユーティリティデータベースへの接続 | デフォルトでは、ユーティリティデータベースへの接続は許可されていません。-su サーバーオプションを使用して、ユーティリティデータベースに接続するパスワードを指定する必要があります。「-su dbeng12/dbsrv12 サーバーオプション」276 ページを参照してください。 | デフォルトでは、ユーティリティデータベースへの接続は、ユーザー ID DBA とパスワードで許可されています。 |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 機能 | ネットワークデータベースサーバー (dbsrv12) | パーソナルデータベースサーバー (dbeng12) |
|---|---|---|
| 接続タイプ | デフォルトで、共有メモリと TCP/IP。 | デフォルトで共有メモリ。TCP/IP プロトコルに接続するには、-x オプションを使用します。「-x dbeng12/dbsrv12 サーバーオプション」289 ページを参照してください。 |
| データベースミラーリング | サポートされています。「データベースミラーリング」1137 ページを参照してください。 | サポートされていません。 |
| LDAP | TCP/IP のみと組み合わせて使用できます。 「LDAP サーバーを使用した接続」103 ページを参照してください。 | サポートされていません。 |
| マルチプログラミングレベルチューニング | サポートされています。「マルチプログラミングレベルのデータベースサーバー設定」165 ページを参照してください。 | サポートされていません。 |
| ネットワーク通信 | サポートされています。 | サポートされていません。 |
| 接続数 | ライセンスで制限されています。「エディションとライセンス」『SQL Anywhere 12 紹介』を参照してください。 | 最大同時接続数は 10 です。 |
| CPU 数 | ライセンスで制限されています。 | 1 です。 |
| 読み込み専用のスケールアウト | サポートされています。「SQL Anywhere の読み込み専用のスケールアウト」1188 ページを参照してください。 | サポートされていません。 |
| Windows Mobile | サポートされています。「SQL Anywhere for Windows Mobile」403 ページを参照してください。 | サポートされていません。 |

### サーバーとデータベースのオプション

次のサーバーとデータベースのオプションは、ネットワークデータベースサーバー**のみ**によってサポートされています。

- 「-gna dbsrv12 サーバーオプション」
- 「-gnh dbsrv12 サーバーオプション」
- 「-gnl dbsrv12 サーバーオプション」
- 「-gns dbsrv12 サーバーオプション」
- 「-pc dbsrv12 サーバーオプション」
- 「-pt dbsrv12 サーバーオプション」
- 「-sm dbsrv12 データベースオプション (旧式)」
- 「-sn dbsrv12 オプション」
- 「-xa dbsrv12 サーバーオプション」
- 「-xf dbsrv12 サーバーオプション」
- 「-xp dbsrv12 データベースオプション」

### データベースサーバープロパティ

次のデータベースサーバープロパティは、ネットワークデータベースサーバー**のみ**によってサポートされています。

- AutoMultiProgrammingLevel サーバープロパティ
- AutoMultiProgrammingLevelStatistics サーバープロパティ
- MaxMultiProgrammingLevel サーバープロパティ
- MinMultiProgrammingLevel サーバープロパティ
- UniqueClientAddresses サーバープロパティ

## データベースサーバーの起動

データベースの開始方法は、使用するオペレーティングシステムによって若干違いがあります。また、コマンドは、使用しているオペレーティングシステムに応じて、複数の方法で指定できます。

- コマンドプロンプトでコマンドを実行する。
- コマンドをショートカットまたはデスクトップアイコンに配置する。
- バッチファイルでコマンドを実行する。
- コマンドを StartLine (START) 接続パラメーターとして接続文字列に含める。「StartLine (START) 接続パラメーター」366 ページを参照してください。

基本的なコマンドの指定方法は、プラットフォームによって若干違いがあります。

**注意**

- 特に指定のないかぎり、これらのコマンドは、パーソナルサーバー (**dbeng12**) を起動します。ネットワークサーバーを起動する場合は、**dbeng12** を **dbsrv12** に置き換えてください。
- データベースファイルがコマンドを開始するディレクトリに含まれている場合は、*path* を指定する必要はありません。
- *database-file* にファイルの拡張子を指定しないと、拡張子は *.db* であると見なされます。

**参照**


- 「SQL Anywhere データベースサーバーの構文」189 ページ
- 「ネットワークプロトコルオプション」369 ページ
- 「SQL スクリプトファイル」『SQL Anywhere サーバー SQL の使用法』
- 「データベースサーバー起動時のトラブルシューティング」1211 ページ


## データベースサーバーの起動例

| コマンド | コメント | その他の情報 |
|---|---|---|
| Windows の場合、**[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[SQL Anywhere]** » **[パーソナルサーバー]** をクリックする。 | **[サーバー起動オプション]** ウィンドウが表示され、パーソナルデータベースサーバー (dbeng12 実行プログラム) を起動するための情報を指定できます。データベースサーバーでデータベースを起動することもできます。 | |
| Windows の場合、**[スタート]** » **[プログラム]** » **[SQL Anywhere]12** » **[SQL Anywhere]** » **[ネットワークサーバー]** をクリックする。 | **[サーバー起動オプション]** ウィンドウが表示され、ネットワークデータベースサーバー (dbsrv12 実行プログラム) を起動するための情報を指定できます。データベースサーバーでデータベースを起動することもできます。 | |

| コマンド | コメント | その他の情報 |
|---|---|---|
| dbeng12 demo | 接続文字列でデータベースファイル名を使用する。<br><br>*demo.db* があるディレクトリでこのコマンドを実行して、パーソナルサーバーと *demo.db* という名前のデータベースサーバーを起動します。 | 「組み込みデータベースの接続」 |
| dbeng12 path/database-file | UNIX | |
| dbeng12 path¥database-file | このコマンドは、Windows Mobile 上でデータベースサーバーを起動する。<br><br>データベースファイルを省略した場合は、**[参照]** をクリックしてデータベースファイルを検索できる **[サーバー起動オプション]** ウィンドウが表示されます。 | 「Windows Mobile デバイスで実行中のデータベースへの接続」 |

# データベースサーバーの停止

データベースサーバーは、次のいずれかの方法で停止します。

●データベースサーバーメッセージウィンドウで **[シャットダウン]** をクリックする。

●dbstop ユーティリティを使用する。

dbstop ユーティリティは、バッチファイルの中で使用するか、別のコンピューターのサーバーを停止するときに便利です。これはコマンドに接続文字列が必要です。「サーバー停止ユーティリティ (dbstop)」1024 ページを参照してください。

●アプリケーションの切断時にデフォルトで自動停止するようにする。

●UNIX でデータベースサーバーメッセージウィンドウが前面に表示されているときに [Q] キーを押す。

**♦ dbstop ユーティリティを使用してデータベースサーバーを停止するには、次の手順に従います。**

1. データベースサーバーを起動します。たとえば、SQL Anywhere インストールディレクトリから実行される次のコマンドは、サンプルデータベースを使用する Ottawa という名前のサーバーを起動します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
dbsrv12 -n Ottawa "%SQLANYSAMP12%¥demo.db"
```

2. dbstop を使用してサーバーを停止します。

```
dbstop -c "Server=Ottawa;UID=DBA;PWD=sql"
```

## データベースサーバーを停止できるユーザー

データベースサーバーの起動時に -gk オプションを使用すると、ユーザーが dbstop を使用してサーバーを停止するために必要なパーミッションレベルを設定できます。パーソナルデータベースサーバーの場合、デフォルト設定は all です。ネットワークデータベースサーバーの場合、必要なパーミッションのデフォルトレベルは DBA ですが、この値を all または none に設定することもできます (ただし、サーバーコンピューターを使用すれば、誰でもデータベースサーバーメッセージウィンドウで **[シャットダウン]** をクリックできます)。

**参照**

- 「-gk dbeng12/dbsrv12 サーバーオプション」235 ページ

## オペレーティングシステムセッションのシャットダウン

データベースサーバーが実行中のオペレーティングシステムセッションを閉じたり、オペレーティングシステムコマンドを使用してデータベースサーバーを停止すると、サーバーは正しく停止しません。そのような操作をすると、次回データベースをロードしたときにリカバリが必要になります。このリカバリは自動的に行われます。

リカバリの詳細については、「バックアップとデータリカバリ」1071 ページを参照してください。

データベースサーバーを明示的に停止してから、オペレーティングシステムのセッションを閉じることをおすすめします。

次に、サーバーを正しく停止しないコマンド例を示します。

- Windows の [タスク マネージャー] でプロセスを停止する
- UNIX の slay または kill コマンドを使用する

# データベースの開始と停止

データベースサーバーは、一度に複数のデータベースをロードできます。次のコマンドで、データベースとデータベースサーバーを同時に起動できます。

```
dbeng12 demo sample
```

**警告**
データベースファイルは、データベースサーバーと同じコンピューター上に置いてください。ネットワークドライブにあるデータベースファイルを操作すると、ファイルが破損することがあります。

### 起動中のサーバー上でのデータベースの開始

次のいずれかの方法で、サーバーの起動後にもデータベースを起動できます。

- サーバーに接続されているときに、DatabaseFile (DBF) 接続パラメーターを使用してデータベースに接続する。DatabaseFile (DBF) 接続パラメーターは、新規接続用のデータベースファイルを指定します。そのデータベースファイルが現在のサーバー上で起動します。

  「組み込みデータベースの接続」28 ページまたは「DatabaseFile (DBF) 接続パラメーター」333 ページを参照してください。

- START DATABASE 文を使用する。

- Sybase Central でサーバーを選択し、**[ファイル]** » **[データベースの開始]** を選択します。

  「START DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

### 制限事項

- サーバーは、固定サイズのページを使用して、データベース情報をメモリに保持します。サーバーがいったん起動すると、サーバーより大きいページサイズのデータベースを起動することはできません。

  「最大ページサイズの考慮事項」150 ページを参照してください。

- -gd サーバーオプションは、データベースの開始に必要なパーミッションを決定します。

## データベースの起動

**♦ 接続しないでデータベースサーバー上でデータベースを起動するには、次の手順に従います (Sybase Central の場合)。**

1. データベースサーバーを選択し、**[ファイル]** » **[データベースの開始]** をクリックします。
2. **[データベースの開始]** ウィンドウに必要な値を入力します。

   データベースが、データベースサーバーの下に未接続のデータベースとして表示されます。

**♦ 接続しないでサーバー上でデータベースを起動するには、次の手順に従います (SQL の場合)。**

Interactive SQL では、データベースに接続して START DATABASE 文を実行する必要があります。

- START DATABASE 文を実行します。

  sample データベースサーバー上で、データベースファイル *c:¥temp¥temp.db* を起動します。

  ```
  START DATABASE 'c:¥¥temp¥¥temp.db'
  AS tempdb ON 'sample'
  AUTOSTOP OFF;
  ```

  AUTOSTOP OFF 接続パラメーターを使用すると、データベースはすべての接続が切断されても自動的には停止されません。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**参照**

- 「START DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「SQL Anywhere データベースサーバーの使用」141 ページ

## データベースの停止

データベースは、次のいずれかの方法で停止します。

- 接続文字列で、起動したデータベースとの接続を切断する。AutoStop (ASTOP) 接続パラメーターを明示的に NO に設定しないかぎり、この動作は自動的に行われます。

  「AutoStop (ASTOP) 接続パラメーター」324 ページを参照してください。

- Interactive SQL または Embedded SQL で STOP DATABASE 文を使用する。

  「STOP DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

データベースの停止の詳細については、「SQL Anywhere データベースサーバーの使用」141 ページを参照してください。

**♦ 切断後にデータベースサーバー上のデータベースを停止するには、次の手順に従います (Sybase Central の場合)。**

現在接続しているデータベースは停止できません。先にデータベースから切断してから、停止する必要があります。データベースを停止するには、同じデータベースサーバー上の別のデータベースに接続されている必要があります。

1. 同じデータベースサーバーにある 1 つ以上の別のデータベースに接続していることを確認します。データベースサーバー上でほかに実行されているデータベースがない場合は、ユーティリティデータベースに接続できます。

2. 停止するデータベースを選択し、**[ファイル]** » **[データベースの停止]** をクリックします。

   データベースから切断すると、そのデータベースが左ウィンドウ枠に表示されなくなることがあります。現在の接続が残っている唯一の接続で、かつデータベースの起動時に AutoStop を指定した場合は、このような状況になります。AutoStop を指定すると、最後の接続が切断されるときに、データベースが自動的に停止します。

**♦ 切断後にサーバー上のデータベースを停止するには、次の手順に従います (SQL の場合)。**

現在接続しているデータベースは停止できません。先にデータベースから切断してから、停止する必要があります。データベースを停止するには、同じデータベースサーバー上の別のデータベースに接続されている必要があります。

1. サーバー上のどのデータベースにも接続していない場合は、ユーティリティデータベースなどのデータベースに接続します。

2. STOP DATABASE 文を実行します。

**参照**

- 「ユーティリティデータベース接続」137 ページ

**例**

次の文は、ユーティリティデータベースに接続して、tempdb データベースを停止します。

```
CONNECT to 'TestEng' DATABASE utility_db
AS conn2
USER 'DBA'
IDENTIFIED BY 'sql';
STOP DATABASE tempdb;
```

別のデータベースを停止するには、データベースに接続されている必要があります。

# データベースサーバー設定

- 各種のオプションを選択して、キャッシュに使用するメモリ量、使用する CPU の数 (ネットワークデータベースサーバーを実行中のマルチプロセッサーコンピューター上)、使用するネットワークプロトコル (ネットワークサーバーのみ) などを指定できます。オプションは、SQL Anywhere の動作とパフォーマンスをチューニングする主要な方法の 1 つです。「SQL Anywhere データベースサーバーの構文」189 ページを参照してください。

- サーバーは、Windows のサービスとして実行できます。サーバーをサービスとして実行すると、コンピューターをログオフしてもサーバーは稼働し続けます。「現在のセッション外でのデータベースサーバーの起動」170 ページを参照してください。

- パーソナルサーバーは、アプリケーションから起動し、アプリケーションの終了と同時にシャットダウンできます。これは、データベースサーバーを組み込みデータベースとして使用する場合に一般的な設定です。「組み込みデータベースの接続」28 ページを参照してください。

**参照**

- 「データベースサーバーの起動」144 ページ

# 最大ページサイズの考慮事項

データベースサーバーキャッシュは、固定サイズのメモリ領域である「ページ」に配置されます。データベースサーバーは停止するまで 1 つのキャッシュを使用するので、ページはすべて同じサイズでなければなりません。

データベースファイルも、データベースの作成時に指定されたサイズのページに配置されます。どのデータベースページも、キャッシュページに適合していなければなりません。デフォルトでは、データベースサーバーページサイズは、データベースサーバーの起動時に指定されたデータベースの最大ページサイズと同じ大きさです。データベースサーバーがいったん起動すると、データベースサーバーページより大きいページサイズのデータベースを起動することはできません。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

サーバーの起動後に大きなページサイズを持つデータベースを起動するには、-gp オプションでページサイズを指定してデータベースサーバーを起動します。より大きいページサイズを使用する場合は、必ずキャッシュサイズを増やしてください。キャッシュサイズを変更しないと、大きなページの一部だけが保管され、領域調整の柔軟性が低くなります。

次のコマンドで、64 MB のキャッシュを予約し、最大 8192 バイトのページサイズを使用するデータベースを収容できるデータベースサーバーを起動します。

```
dbsrv12 -gp 8192 -c 64M -n myserver
```

**参照**

- 「-gp dbeng12/dbsrv12 サーバーオプション」242 ページ

# データベースサーバー名とデータベース名を指定する

-n は、データベースサーバーオプション (データベースサーバーの指定) またはデータベースオプション (データベースの指定) として使用できます。

クライアントアプリケーションは、データベースへの接続時にデータベースサーバー名とデータベース名を指定する接続パラメーターを使用できます。データベースサーバー名は、デスクトップアイコンとデータベースサーバーメッセージウィンドウのタイトルバーに表示されます。

utility_db という名前を使用してデータベースを作成したり、データベースサーバーを起動したりすることはできません。これは、この名前が SQL Anywhere ユーティリティデータベース用に予約されているためです。「ユーティリティデータベース」135 ページを参照してください。

### データベースサーバーの命名

データベースサーバーに名前を付けると、ネットワーク上の他のデータベースサーバー名との重複を防ぐことができます。また、クライアントアプリケーションのユーザーにわかりやすい名前を提供できます。データベースサーバーは、停止するまでその名前を維持します。データベースサーバー名を指定しない場合は、最初に起動されたデータベースの名前になります。

最初のデータベースファイルの前に -n オプションを指定すると、データベースサーバーに名前を付けることができます。たとえば、次のコマンドは、サンプルデータベースを実行するデータベースサーバーを起動し、そのサーバーに Cambridge という名前を付けます。

```
dbeng12 -n Cambridge "%SQLANYSAMP12%¥demo.db"
```

データベースサーバー名を指定すると、データベースを起動せずにデータベースサーバーを起動できます。次のコマンドは、データベースを起動せずに Galt という名前のデータベースサーバーを起動します。

```
dbeng12 -n Galt
```

データベースサーバーの名前は、有効な識別子であることが必要です。データベースサーバーのロングネームは、プロトコルごとに異なる長さにトランケートされます。データベースサーバー名には、次に該当する値を指定できません。

- 空白スペース、一重引用符、または二重引用符で始まる値
- 空白スペースで終わる値
- セミコロン、フォワードスラッシュ (/)、またはバックスラッシュ (¥) を含む値
- 長さが 250 バイトを超える値
- UNIX 上で実行するときにスペースを含む値

実行中のサーバーでのデータベースの起動については、「データベースの開始と停止」147 ページを参照してください。

**注意**
Windows と UNIX では、データベースサーバーがバージョン 10.0.0 以降で、名前が次の長さを超えている場合、バージョン 9.0.2 以前のクライアントから接続することはできません。

- Windows 共有メモリの場合は、40 バイト
- UNIX 共有メモリの場合は、31 バイト
- TCP/IP の場合は、40 バイト

### データベースの命名

また、クライアントアプリケーションのユーザーにわかりやすいデータベースの名前を提供できます。データベースは、停止されるまでその名前で識別されます。データベース名の最大長は 250 バイトです。

データベース名を指定しない場合、データベースファイル名のルート (*.db* 拡張子を省いたファイル名) がデフォルトのデータベース名になります。たとえば、次のコマンドでは、最初のデータベース名は mydata、次のデータベース名は mysales です。

```
dbeng12 c:¥mydata.db c:¥sales¥mysales.db
```

データベースファイルの後に -n オプションを指定すると、データベースに名前を付けることができます。たとえば、次のコマンドはサンプルデータベースを起動し、それに MyDB という名前を付けます。

```
dbeng12 "%SQLANYSAMP12%¥demo.db" -n MyDB
```

### 大文字と小文字の区別

データベースサーバー名とデータベース名は、文字セットがシングルバイトの場合は大文字と小文字が区別されません。「接続文字列と文字セット」478 ページを参照してください。

## 設定ファイルとデータベースサーバーの起動オプション

設定ファイルにデータベースサーバーの起動に使用する一連のオプションを格納し、そのファイルをデータベースサーバーコマンドで呼び出すことができます。設定ファイルには、複数行にわたってオプションを保存できます。たとえば、次の設定ファイルは、パーソナルデータベース

サーバーとサンプルデータベースを起動します。また、キャッシュを 10 MB に設定し、パーソナルサーバーのこのインスタンスの名前を **Elora** にします。# で始まる行は、コメントとして処理されます。

```
# Configuration file for server Elora
-n Elora
-c 10M
"%SQLANY12%¥demo.db"
```

これらのオプションを含むファイルに *sample.cfg* という名前を付けた場合は、このファイルを次のように指定できます。

```
dbeng12 @sample.cfg
```

**参照**


- 「@data dbeng12/dbsrv12 サーバーオプション」201 ページ
- 「設定ファイル」909 ページ
- 「設定ファイルでの条件付き解析」910 ページ


# データベースサーバー動作のロギング

「データベースサーバーメッセージログ」には、情報メッセージ、エラー、警告、MESSAGE 文からのメッセージが含まれています。これらのメッセージは、次の場所に表示されます。

- データベースサーバーメッセージウィンドウ (Windows の場合はシステムトレイアイコン)
- Sybase Central の **[サーバーメッセージと実行された SQL]** ウィンドウ枠
- SQL Anywhere コンソールユーティリティ
- データベースサーバーメッセージログファイル
- データベースサーバーをコマンドラインアプリケーションとして実行した場合のコマンドプロンプトウィンドウまたはシェル
- UNIX Syslog

**参照**


- 「-o dbeng12/dbsrv12 サーバーオプション」258 ページ
- 「-oe dbeng12/dbsrv12 サーバーオプション」259 ページ
- 「-on dbeng12/dbsrv12 サーバーオプション」260 ページ
- 「-os dbeng12/dbsrv12 サーバーオプション」261 ページ
- 「-ot dbeng12/dbsrv12 サーバーオプション」262 ページ


## ファイルへのデータベースサーバーメッセージのロギング

デフォルトでは、データベースサーバーメッセージはデータベースサーバーメッセージウィンドウに送信されます。-o オプションを使用すると、結果をログファイルに送信できます。次のコマンドを実行して、結果を *mydbserver_messages.txt* という名前のログファイルに送ります。

```
dbsrv12 -o mydbserver_messages.txt -c ...
```

データベースサーバーメッセージログファイルのサイズを制御したり、ファイルが最大サイズに達したときの処理を指定したりできます。

- -o オプションを使用して、データベースサーバーメッセージログファイルを使用することを指定し、ファイル名を入力します。

- -ot オプションを使用して、データベースサーバーメッセージログファイルを使用することを指定し、ファイル名を入力すると、新しいメッセージが送信される前にログファイルの前の内容が削除されます。

- -o または -ot に加えて -on オプションを使用してサイズを指定すると、そのサイズに達したときに、これまでのデータベースサーバーメッセージログファイルの名前に拡張子 *.old* が付けられて変更され、元の名前を持つ新しいファイルが使用されます。

- -o または -ot に加えて -os オプションを使用してサイズを指定すると、そのサイズに達したときに、日付と連番に基づいた新しい名前を持つ新しいデータベースサーバーメッセージログファイルが使用されます。

起動エラー、致命的なエラー、アサーションのロギング先をそれぞれ別ファイルにするには、-oe オプションを指定します。

トランザクションログを使用した操作を実行するユーティリティで問題が発生する場合があるため、データベースサーバーメッセージログファイル名の最後には *.log* を付けないようにしてください。

**参照**


- 「-o dbeng12/dbsrv12 サーバーオプション」258 ページ
- 「-oe dbeng12/dbsrv12 サーバーオプション」259 ページ
- 「-on dbeng12/dbsrv12 サーバーオプション」260 ページ
- 「-os dbeng12/dbsrv12 サーバーオプション」261 ページ
- 「-ot dbeng12/dbsrv12 サーバーオプション」262 ページ


## Sybase Central での SQL 文のロギング

Sybase Central でデータベースを編集する場合、アクションに応じて自動的に SQL 文が生成されます。これらの文は **[サーバーメッセージと実行された SQL]** という別のウィンドウ枠で追跡できます。または情報をファイルに保存できます。**[サーバーメッセージと実行された SQL]** ウィンドウ枠には、データベースとデータベースサーバーごとにタブがあります。データベースサーバーのタブには、データベースサーバーメッセージウィンドウと同じ情報が表示されます。

Interactive SQL の場合、実行した文のログを取ることもできます。「Interactive SQL での文のロギング」856 ページを参照してください。

**♦ [サーバーメッセージと実行された SQL] ウィンドウ枠に、Sybase Central によって生成された SQL 文のログを取るには、次の手順に従います。**

1. **SQL Anywhere 12** プラグインを使用して、データベースに接続します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

2. **[表示]** » **[サーバーメッセージと実行された SQL]** をクリックします。

3. **[サーバーメッセージと実行された SQL]** ウィンドウ枠で、データベースアイコンがあるタブをクリックします。

4. 右クリックして **[オプション]** をクリックします。

5. ロギングのオプションを編集します。

6. **[保存]** をクリックします。

**♦ ファイルに、Sybase Central によって生成された SQL 文のログを取るには、次の手順に従います (Sybase Central の場合)。**

1. データベースに接続します。

2. 左ウィンドウ枠で、データベースを右クリックし、**[データベース変更のロギングの開始]** をクリックします。

3. ファイル名を指定し、**[保存]** をクリックします。

**♦ データベース変更のロギングを停止するには、次の手順に従います (Sybase Central の場合)。**

● 左ウィンドウ枠で、データベースを右クリックし、**[データベース変更のロギングの停止]** を選択します。

# Windows イベントログメッセージの抑制

レジストリエントリを設定することによって、Windows イベントログのエントリを抑制できます。レジストリエントリは *Software¥Sybase¥SQL Anywhere¥12.0* です。このエントリは、HKEY_CURRENT_USER または HKEY_LOCAL_MACHINE ハイブのいずれかに配置できます。

イベントログエントリを制御するには、種類が REG_DWORD の EventLogMask キーを設定します。この値はビットマスクで、さまざまなイベントメッセージに関する内部ビット値を保有します。

```
errors      EVENTLOG_ERROR_TYPE       0x0001
warnings    EVENTLOG_WARNING_TYPE     0x0002
information EVENTLOG_INFORMATION_TYPE 0x0004
```

たとえば、EventLogMask キーを 0 に設定すると、どのメッセージも出力されなくなります。このキーを 1 に設定すると、情報メッセージと警告メッセージは出力されませんが、エラーメッセージは出力されます。デフォルト設定 (エントリが存在しない場合) では、すべてのメッセージタイプが出力されます。

EventLogMask キーの設定を変更した場合、変更を有効にするには、データベースサーバーを再起動する必要があります。

**参照**

- 「ネットワークプロトコルオプション」369 ページ

# コマンドラインからパフォーマンスとメモリを制御する

次のようないくつかの設定は、データベースサーバーのパフォーマンスに影響することがあります。

- **キャッシュサイズ**　データベースサーバーに割り当て可能なキャッシュメモリの容量は、パフォーマンスに影響を及ぼす主要な要因の 1 つになります。データベースサーバーが利用できるメモリが多いほど、実行速度が速くなります。キャッシュには何度も要求される情報が保持されます。ディスクの情報にアクセスするよりも、キャッシュ内の情報にアクセスする方が速くなります。デフォルトの初期キャッシュサイズは、物理メモリの容量、オペレーティングシステムとデータベースファイルのサイズに基づいて計算されます。データベースサーバーは、必要に応じてキャッシュサイズを自動的に調整します。「動的キャッシュサイズ決定」『SQL Anywhere サーバー SQL の使用法』を参照してください。

  データベースサーバーメッセージウィンドウには起動時のキャッシュサイズが表示されます。また、次の文を使用して現在のキャッシュサイズを取得することもできます。

  ```
  SELECT PROPERTY( 'CurrentCacheSize' );
  ```

  パフォーマンスチューニングの詳細については、「パフォーマンスのモニタリングと診断のツール」『SQL Anywhere サーバー SQL の使用法』を参照してください。

  次の表は、キャッシュの制御に使用できるデータベースサーバーオプションをまとめたものです。

| キャッシュ機能 | データベースサーバーオプション | 用途 | 参照 |
|---|---|---|---|
| キャッシュサイズ | -c | データベースサーバーキャッシュ用のメモリの初期容量を設定します。 | 「-c dbeng12/dbsrv12 サーバーオプション」204 ページ |
| | -ca 0 | 静的キャッシュサイズを強制的に適用します。 | 「-ca dbeng12/dbsrv12 サーバーオプション」206 ページ |
| | -ch | 自動キャッシュサイズ変更の最大キャッシュサイズを設定します。 | 「-ch dbeng12/dbsrv12 サーバーオプション」208 ページ |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| キャッシュ機能 | データベースサーバーオプション | 用途 | 参照 |
|---|---|---|---|
| | -chx | キャッシュ以外の用途のアドレス領域を予約しないで自動キャッシュサイズ変更の最大キャッシュサイズを設定します(32 ビットデータベースサーバーの場合のみ)。 | 「-chx dbeng12/dbsrv12 サーバーオプション」209 ページ |
| | -cl | 自動キャッシュサイズ変更の最小キャッシュサイズを設定します。 | 「-cl dbeng12/dbsrv12 サーバーオプション」211 ページ |
| | -cs | データベースサーバーメッセージウィンドウに、動的なキャッシュサイズ変更に関する統計を表示します。 | 「-cs dbeng12/dbsrv12 サーバーオプション」215 ページ |
| キャッシュウォーミング | -cc | 次回にデータベースが起動されるときに、キャッシュウォーミングに使用するデータベースページに関する情報を収集します。 | 「-cc dbeng12/dbsrv12 サーバーオプション」207 ページ |
| | -cr | データベースページを保持するキャッシュを準備します。 | 「-cr dbeng12/dbsrv12 サーバーオプション」214 ページ |
| | -cv | データベースサーバーメッセージウィンドウに、キャッシュウォーミングに関するメッセージを表示します。 | 「-cv dbeng12/dbsrv12 サーバーオプション」216 ページ |

| キャッシュ機能 | データベースサーバーオプション | 用途 | 参照 |
|---|---|---|---|
| Address Windowing Extensions (AWE) キャッシュ (旧式) | -cm | Windows において、AWE キャッシュに割り付けるアドレス領域のサイズを設定します。 | 「-cm dbeng12/dbsrv12 サーバーオプション」212 ページ |
| | -cw | Windows での AWE の使用を有効にします。 | 「-cw dbeng12/dbsrv12 サーバーオプション (旧式)」217 ページ |

**注意**
AWE は使用されなくなりました。サイズの大きいキャッシュが必要な場合は、64 ビット版の Windows オペレーティングシステムで 64 ビット版の SQL Anywhere データベースサーバーを使用することをおすすめします。

- **マルチプログラミングレベル**　データベースサーバーのマルチプログラミングレベルは、同時に実行できるデータベースサーバータスクの最大数を指定するものです。一般に、マルチプログラミングレベルを上げるほど、同時に実行する要求の数が増えるため、データベースサーバーの全体的なスループットは向上します。ただし、マルチプログラミングレベルを上げすぎると、同じリソースを使用する要求が複数存在するときに、余分な競合が発生し、トランザクションの応答時間が長くなります。

  デフォルトでは、SQL Anywhere はデータベースサーバーのマルチプログラミングレベルを自動的に調整します。マルチプログラミングレベルを上げると、場合によっては、システムのスループットが低下することもあります。次のオプションを使用すると、データベースサーバーのマルチプログラミングレベルを手動で制御できます。

| データベースサーバーオプション | sa_server_option 値 | 説明 |
|---|---|---|
| 「-gn dbsrv12 サーバーオプション」237 ページ | CurrentMultiProgrammingLevel | データベースサーバーのマルチプログラミングレベルを設定します。 |
| 「-gna dbsrv12 サーバーオプション」238 ページ | AutoMultiProgrammingLevel | データベースサーバーのマルチプログラミングレベルの動的チューニングをオンまたはオフにします。 |
| 「-gnh dbsrv12 サーバーオプション」239 ページ | MaxMultiprogrammingLevel | データベースサーバーが同時に実行できるタスクの最大数を設定します。 |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| データベースサーバーオプション | sa_server_option 値 | 説明 |
|---|---|---|
| 「-gnl dbsrv12 サーバーオプション」240 ページ | MinMultiProgrammingLevel | データベースサーバーが同時に実行できるタスクの最小数を設定します。 |
| 「-gns dbsrv12 サーバーオプション」241 ページ | AutoMultiProgrammingLevelStatistics | マルチプログラミングレベルの自動変更に関する統計をデータベースサーバーメッセージログに出力するかどうかを制御します。<br><br>sa_server_option システムプロシージャーで AutoMultiProgrammingLevelStatistics プロパティを使用して、この動作を制御することもできます。「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |

SQL Anywhere のマルチプログラミングレベルの詳細については、「マルチプログラミングレベルのデータベースサーバー設定」165 ページを参照してください。

- **プロセッサーの数**　ネットワークデータベースサーバーを使用するマルチプロセッサーコンピューターを実行している場合、-gt オプションを使用してプロセッサー数を設定できます。「-gt dbeng12/dbsrv12 サーバーオプション」244 ページと「SQL Anywhere のスレッド化」161 ページを参照してください。

  データベースサーバーで使用できる CPU 数には、ライセンスまたは SQL Anywhere のエディションも影響する場合があります。「エディションとライセンス」『SQL Anywhere 12 紹介』を参照してください。

- **その他のパフォーマンスに関連するオプション**　ネットワークのパフォーマンスをチューニングするオプションには、-gb (データベース処理優先度) と -u (バッファーディスク I/O) などいくつかのオプションがあります。「SQL Anywhere データベースサーバーの構文」189 ページを参照してください。

# コマンドラインからパーミッションを制御する

ある特定のグローバルオペレーションの実行に必要なパーミッションを制御するオプションがあります。制御されるパーミッションには、データベースの開始と停止、データのロードとアン

ロード、データベースファイルの作成と削除などを行うものが含まれます。「データベースサーバーの保護」1395 ページを参照してください。

# 特殊モード

SQL Anywhere を特殊モードで実行できます。

- **読み込み専用**　-r オプションを入力すると、データベースを読み込み専用モードで起動できます。監査がオンになっているデータベースを読み込み専用モードで起動することはできません。「-r dbeng12/dbsrv12 サーバーオプション」267 ページと「-r dbeng12/dbsrv12 データベースオプション」313 ページを参照してください。

- **インメモリモード**　-im オプションを指定すると、データベースを完全にインメモリで実行できます。チェックポイントモードのみ (-im c) で実行すると、データベースサーバーはトランザクションログを使用しませんが、最新のチェックポイントにリカバリできます。非書き込みモード (-im nw) でデータベースを実行すると、コミットされたトランザクションはディスク上のデータベースファイルに書き込まれないため、データベースを停止すると、すべての変更は失われます。どちらのインメモリモードを使用した場合も、データベースがアクティブである間は、アプリケーションから変更を加えたり、アクセスしたりできます。「-im dbeng12/dbsrv12 サーバーオプション」247 ページを参照してください。

  **注意**
  インメモリモードには別途ライセンスが必要です。「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

- **バルクロード**　これは、Interactive SQL の INPUT 文を使用してデータベースに大量のデータをロードするときに便利です。LOAD TABLE を使用してデータをバルクロードする場合は、-b オプションは使用しないでください。「-b dbeng12/dbsrv12 サーバーオプション」203 ページと「データのインポートとエクスポート」『SQL Anywhere サーバー SQL の使用法』を参照してください。

- **トランザクションログなしの起動**　-f データベースオプションは、リカバリ時、つまりトランザクションログの消失後にデータベースサーバーを起動したり、トランザクションログが見つからないときにデータベースサーバーを起動したりする場合に使用します。-f オプションはサーバーオプションではなく、データベースオプションです。

  リカバリが完了したら、サーバーを停止し、-f オプションを指定せずに再起動してください。「-f dbeng12/dbsrv12 サーバーリカバリオプション」228 ページを参照してください。

- **クワイエットモードで作動する**　データベースサーバーはクワイエットモードをサポートしています。サーバーをどのようなクワイエットモードで操作するかを決定します。クワイエットモードの範囲には、メッセージやシステムトレイのアイコンを非表示にするモードから、完全に非表示にするモードまであります。Windows 上のデータベースサーバーを完全に非表示にするモードで操作するには、-qi、-qs、-qw オプションを指定します。これらのオプションを設定すると、すべてのアイコンとすべての起動エラーメッセージが表示されなくなるので、サーバーが実行中であることを視覚的に表示するものがなくなります。データベー


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

スサーバーをクワイエットモードで実行する場合、-o または -oe オプションのいずれか (または両方) を使用してエラーを診断できます。

-qi と -qs オプションを使用しても、-v (バージョン) と -ep (データベース暗号化パスワードを要求) サーバーオプションによるウィンドウは表示されます。次の項を参照してください。

○「-qi dbeng12/dbsrv12 サーバーオプション」264 ページ
○「-qn dbeng12/dbsrv12 サーバーオプション」265 ページ
○「-qs dbeng12/dbsrv12 サーバーオプション」266 ページ
○「-qw dbeng12/dbsrv12 サーバーオプション」267 ページ

# SQL Anywhere のスレッド化

SQL Anywhere のスレッドモデルを理解するには、スレッドと要求処理の基本用語と概念を理解することが必要です。

- **要求**　要求は、クエリや SQL 文などの作業単位で、接続を介してデータベースサーバーに送信されます。要求の存続期間は、要求がデータベースによって最初に受信されてから、結果の最後が返されてカーソルが閉じられるまで、または要求がキャンセルされるまでの間を指します。

- **タスク**　タスクは、データベースサーバー内で実行されるアクティビティの単位です。また、データベースサーバーによってスケジュールされる最小の作業単位でもあります。ユーザー要求は、それぞれ、データベースサーバー内で少なくとも 1 つのタスクになります。クエリ内並列処理が関連する場合は、複数のタスクになることもあります。データベースサーバーは、ユーザー要求だけでなく、内部タスクを実行するために自身のタスクをスケジュールすることもできます。たとえば、データベースクリーナーの実行やタイマーの処理などがこれに該当します。同時に実行できるアクティブなタスクの最大数は、カーネル内のワーカープールのサイズによって異なります。ユーザー要求タスクがデータベースサーバーに届き、そのタスクにワーカーが割り当てられると、ActiveReq サーバープロパティの値が 1 増えます。要求が完了すると、ActiveReq サーバープロパティの値は 1 減ります。ただし、ユーザー要求タスクを実行するために使用できるワーカーがない場合、そのタスクは実行のためにキューイングされ、UnschReq サーバープロパティの値が 1 増えます。ワーカーが使用可能になり、タスクが実行のためにデキューされると、UnschReq サーバープロパティの値が 1 減り、ActiveReq プロパティの値が 1 増えます。UnschReq サーバープロパティ778 ページとActiveReq サーバープロパティ754 ページを参照してください。

- **ワーカー**　ワーカーは、データベースサーバーカーネル内で実行されている「制御用のスレッド」を抽象化したものです。Windows と Linux では、ワーカーは「ファイバー」と呼ばれる軽量スレッドを使用して実装されます。その他のプラットフォームでは、オペレーティングシステムスレッドを使用して実装されます。ワーカーは、データベースサーバーのカーネルによって割り当てられたタスクを実行します。データベースサーバーは、サイズが可変のワーカープールを利用してサーバーの負荷を処理します。プールのサイズは、データベースサーバーのマルチプログラミングレベルに対応します。

  パーソナルサーバーと Windows Mobile 上のネットワークサーバーには、固定サイズのプールがあります。これらのサーバーでは、サーバーの起動時に作成されるワーカーの数を -gn オ

プションで制御します。他のすべてのネットワークサーバーでは、-gnh オプションで指定された数のワーカーがデータベースサーバーによって作成されます。ただし、タスクに対してサービスを提供できるのは、サーバーのマルチプログラミングレベルで指定された数のワーカーのみです。デフォルトでは、ワーカープールのサイズは、データベースサーバーの負荷の変化に応じてカーネルによって動的にチューニングされ、-gnl オプションと -gnh オプションでそれぞれ指定された下限と上限の間で上下します。「マルチプログラミングレベルのデータベースサーバー設定」165 ページを参照してください。

タスクは、先入れ先出し (FIFO) 方式でワーカーに割り当てられます。各タスクの優先度は、そのタスクを生成した接続に基づいて設定されます。タスクが実行のためにワーカーに割り当てられると、カーネルのスケジューラーはタスクの優先度を考慮して CPU 時間をそのワーカーに割り付けます。タスクの実行中、ロックを待機したり、I/O 処理の完了を待機したりするなどの理由でタスクのブロックが必要になった場合でも、ワーカーはそのタスクに関連付けられたままです。タスクが完了した後にのみ、ワーカーは他のタスクを実行するために使用できるようになります。

- **スレッド**　スレッド、つまり実行スレッドは、オペレーティングシステムを構築するもので、単一プロセス内での同時実行を可能にします。データベースサーバーを含め、オペレーティングシステムの各プロセスは、少なくとも 1 つのスレッドによって実行されます。複数のスレッドによって実行されることもあります。スレッドは、オペレーティングシステムによってアプリケーション外部でスケジュールされ、最終的にアプリケーション実行のすべてはスレッドによって行われます。Windows と Linux では、データベースサーバーは決められた数のスレッドを作成します。つまり、CPU コアごとに 1 つのオペレーティングシステムスレッドを作成し、この数は -gtc オプションで制御されます。その他のプラットフォームでは、作成されるオペレーティングシステムスレッドの数はワーカープールのサイズと同じであり、ワーカープールサイズの制御に使用されるのと同じメカニズムで制御されます。「-gtc dbeng12/dbsrv12 サーバーオプション」245 ページと「-gn dbsrv12 サーバーオプション」237 ページを参照してください。

  スレッドの数は、データベースへの接続の数とは関係ありません。データベースサーバーは、スレッドを特定の接続専用に割り当てません。代わりに、タスクは、実行のためにデータベースサーバーに届くと、サーバースレッドの固定サイズのプールからスレッドが動的に割り当てられます。タスクがスレッドに割り当てられると、スレッドはタスクが完了するかキャンセルされるまでタスクを処理します。

**参照**


- 「スレッド動作」164 ページ
- 「マルチプログラミングレベルのデータベースサーバー設定」165 ページ
- 「-gn dbsrv12 サーバーオプション」237 ページ
- 「-gnl dbsrv12 サーバーオプション」240 ページ
- 「-gnh dbsrv12 サーバーオプション」239 ページ
- 「-gtc dbeng12/dbsrv12 サーバーオプション」245 ページ
- 「sa_clean_database システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「トランザクションのブロックとデッドロック」『SQL Anywhere サーバー SQL の使用法』


 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## UNIX でのワーカー

UNIX では、ワーカーはオペレーティングシステムスレッドを使用して実装されます。したがって、タスクのスケジュールはオペレーティングシステムのスケジューラーによって制御され、オペレーティングシステムはいつでもスレッドの実行を先取りする可能性があります。このプリエンプティブスケジューリングは、タスクの処理に対して、目に見える形で影響を及ぼすことはありません。スレッドが再び実行されるようにスケジュールされると、タスクは中断したところから処理が開始されます。ただし、データベース接続から継承されたタスクの優先度は、それらのタスクの実行スケジュールに影響を及ぼしません。代わりに、すべてのオペレーティングシステムスレッドの優先度が同じであるため、すべてのタスクは同じ優先度レベルで実行されます。

## Windows と Linux でのワーカー

Windows と Linux の場合、ワーカーは「ファイバー」と呼ばれる軽量スレッドを使用して実装されます。ファイバーを利用した実装によって、オペレーティングシステムスレッドのスケジューラーでワーカーを先取りでスケジュールするのではなく、協調的にスケジュールできます。その結果、ファイバー間の切り替えをより詳細に制御できるようになります。データベースカーネルは、プロセッサーの結び付きを最大にして各ワーカーの実行優先度を密接に制御するようにファイバーをスケジュールします。

ファイバーは、オペレーティングシステムスケジューラーに依存しないため、他のアクティビティの完了を待機しているときは、制御を別のファイバーに明示的に譲ります。たとえば、ファイバーで実行中のタスクにおいて、I/O 操作の完了を待機するなどの理由でブロックが必要になると、ファイバーは制御を解放して他のファイバーに譲ります。元のファイバーを実行しているスレッドは、カーネルを介した切り替えを行うことなく、即座に別のファイバーを選択して実行を開始できます。ファイバーがブロックしたが制御を譲らなかった場合、そのファイバーを実行しているスレッドがブロックされ、他のファイバーがそのスレッドで実行できなくなります。複数のスレッドがファイバーを実行している場合、待機中のファイバを実行しているスレッドだけがブロックされます。他のスレッドは、ファイバーを自由に実行できます。

Windows と Linux プラットフォームには、少なくとも、データベースサーバーの最大同時実行性設定 (-gnh オプションで指定) で求められた数と同数のファイバーが存在します。データベースサーバーは、サーバーの実行中に、現在のサーバーマルチプログラミングレベルに対応する、これらのファイバーのサブセットを利用します。ファイバーごとに、オペレーティングシステムはスタックのアドレス領域を予約する必要があります。各ファイバーに必要なスタックのサイズは、-gss サーバーオプションの設定によって異なります。

ネットワークサーバーで必要な実行スタックのアドレス領域のサイズは、最大マルチプログラミングレベルに比例します。

**参照**


- 「-gss dbeng12/dbsrv12 サーバーオプション」243 ページ
- 「-gn dbsrv12 サーバーオプション」237 ページ

## スレッド動作

スレッドの動作は、いくつかの要素によって制御されます。これらは、サーバーオプションによって管理されます。すべてのプラットフォームで、これらのオプションのすべてがサポートされるわけではありません。

- **マルチプログラミングレベル (-gn サーバーオプション)**　-gn オプションは、データベースサーバーのマルチプログラミングレベルを制御します。この値によって、同時にアクティブにできるタスクの最大数が決定されます。データベース要求ごとに、少なくとも 1 つのタスクが作成されます。クエリ内並列処理が関連する場合は、複数のタスクが作成されることもあります。また、サーバーは、内部的な処理を行うために臨時にタスクをスケジュールすることもあります。サーバー内にあるタスクの数がマルチプログラミングレベルを超えると、未処理のタスクは、現在実行中のタスク (アクティブなタスク) が完了するのを待機します。「-gn dbsrv12 サーバーオプション」237 ページと「マルチプログラミングレベルのデータベースサーバー設定」165 ページを参照してください。

- **内部実行スレッドあたりのスタックサイズ (-gss サーバーオプション)**　-gss オプションを使用して、データベースサーバーのワーカーあたりのスタックサイズを設定できます。-gss オプションによって、データベースサーバー内の各ワーカーに必要なアドレス領域を減らすことができます。これは、メモリが限られている環境で役立ちます。「-gss dbeng12/dbsrv12 サーバーオプション」243 ページを参照してください。

- **プロセッサーの数 (-gt サーバーオプション)**　-gt オプションは、データベースサーバーが使用するプロセッサーの数を制御します。「-gt dbeng12/dbsrv12 サーバーオプション」244 ページを参照してください。

- **プロセッサーの同時実行性 (-gtc サーバーオプション)**　-gtc オプションは、データベースサーバーが使用する論理プロセッサー (コア) の数を指定します。「-gtc dbeng12/dbsrv12 サーバーオプション」245 ページを参照してください。

### プロセッサーの使用とスレッド処理の例

次の例は、-gt と -gtc の設定に基づいて、データベースサーバーが CPU を選択する方法を示します。この例では、システムが 4 プロセッサー構成で、各プロセッサーにコアが 2 つあることを前提としています。物理プロセッサーは文字で、コアは数字でそれぞれ区別します。したがって、このシステムにはプロセッサーユニットとして A0、A1、B0、B1、C0、C1、D0、D1 が存在することになります。

| シナリオ | ネットワークデータベースサーバーの設定 |
|---|---|
| シングル CPU ライセンス、または -gt に 1 が指定されている | ● -gt 1<br>● -gtc 2<br><br>スレッドは A0 または A1 で実行できます。 |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| シナリオ | ネットワークデータベースサーバーの設定 |
|---|---|
| CPU のライセンスに制限がなく、-gtc に 5 が指定されている | ● -gt 4<br>● -gtc 5<br><br>スレッドは A0、A1、B0、C0、D0 で実行できます。 |
| データベースサーバーに 3 CPU ライセンスがあり、-gtc に 5 が指定されている | ● -gt 3<br>● -gtc 5<br><br>スレッドは A0、A1、B0、B1、C0 で実行できます。 |
| CPU のライセンスに制限がなく、-gtc に 1 が指定されている | ● -gt 4<br>● -gtc 1<br><br>スレッドは A0 でのみ実行できます。 |

## マルチプログラミングレベルのデータベースサーバー設定

データベースサーバーの「マルチプログラミングレベル」は、同時にアクティブにできるタスクの最大数ですクライアント側の要求がデータベースサーバーに届くと、その要求用に作成されたタスクがワーカー (使用可能な場合) に割り当てられます。ワーカーが割り当てられた要求は「アクティブな要求」と呼ばれます。使用可能なすべてのワーカーがビジー状態の場合、要求はスケジュールされていない要求キューと呼ばれる特殊なキューに格納され、要求はスケジュールされていない要求として分類されます。同様に、アクティブなタスクは、ワーカーによって現在実行されているタスクです。アクティブなタスクは、アクセスプランの演算子を実行したり、その他のファンクションを実行したりします。ただし、(I/O 操作やローロックなどのために) リソースを待機することもあります。スケジュールされていないタスクは、実行の準備が整っているものの、利用可能なワーカーを待機しているタスクを指します。同時に実行できるアクティブなタスクの数は、サーバーのスレッドの数と、コンピューターに搭載されている論理プロセッサーの数によって異なります。

SQL Anywhere を使用すると、DBA は、データベースサーバーがサーバースループットに基づいてマルチプログラミングレベルを動的にチューニングできるようにするか (デフォルト)、またはマルチプログラミングレベルを手動で設定するかを選択できます。マルチプログラミングレベルは、データベースサーバーの起動時にデータベースサーバーオプション (-gna、-gnl、-gnh) を指定して設定するか、またはデータベースサーバーの実行後に sa_server_option システムプロシージャーで MinMultiprogrammingLevel、MaxMultiprogrammingLevel、CurrentMultiprogrammingLevel プロパティを使用して設定できます。

次の表は、データベースサーバーのマルチプログラミングレベルを制御するコマンドラインオプションとサーバーオプションをまとめたものです。

| データベースサーバーオプション (データベースサーバーの起動時) | sa_server_option 値 (データベースサーバーの実行中) | 説明 |
|---|---|---|
| -gn | CurrentMultiProgrammingLevel | データベースサーバーのマルチプログラミングレベルを設定します。 |
| -gna | AutoMultiProgrammingLevel | データベースサーバーのマルチプログラミングレベルの動的チューニングをオンまたはオフにします。 |
| -gnh | MaxMultiprogrammingLevel | ネットワークデータベースサーバーが同時に実行できるタスクの最大数を設定します。 |
| -gnl | MinMultiProgrammingLevel | ネットワークデータベースサーバーが同時に実行できるタスクの最小数を設定します。 |

### マルチプログラミングレベルのチューニング

SQL Anywhere ネットワークサーバーは、サーバーのマルチプログラミングレベルの動的チューニングと手動チューニングの両方をサポートしています。

- **マルチプログラミングレベルの動的チューニング**　SQL Anywhere ネットワークサーバーは、データベースサーバーのスループットレベルを自動的にモニターし、現在の負荷に応じてマルチプログラミングレベルを調整する方法を決定できます。データベースサーバーは山登り法というアルゴリズムと放物線近似法を使用して、必要な調整を決定します。マルチプログラミングレベルを上げるとネットワークデータベースサーバーのスループットレベルが上がる場合、ネットワークデータベースサーバーはそのまま処理を続行します。マルチプログラミングレベルを上げるとスループットが低下する場合、データベースサーバーはマルチプログラミングレベルを下げます。データベースサーバーはスループットレベルを常時モニターし、サーバースループットが向上するようにマルチプログラミングレベルを変更します。多数の要求が短時間に集中し、その後長いアイドル時間が続く場合は、最小マルチプログラミングレベルを予想される最大同時実行レベルに設定することをおすすめします。このように設定すると、データベースサーバーは要求が短時間に集中している間でも応答できます。

  マルチプログラミングレベルの動的チューニングは、デフォルトでは有効になっています (-gna 1)。データベースサーバーオプション -gnl と -gnh は、動的チューニングアルゴリズムで使用されるマルチプログラミングレベルの最小値と最大値をそれぞれ設定します。動的チューニングが有効になっている場合、スレッドデッドロックが発生するたびに、データベースサーバーはマルチプログラミングレベルを自動的に上げてスレッドデッドロックの問題を排除しようとします。データベースサーバーは、許容される MaxMultiprogrammingLevel 値になるまでマルチプログラミングレベルを上げ続けます。MaxMultiprogrammingLevel 値に達すると、データベースサーバーはスレッドデッドロック問題をクライアントアプリケーション

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

に返し始めます。ThreadDeadlocksAvoided サーバープロパティ777 ページを参照してください。

- **マルチプログラミングレベルの手動チューニング**　マルチプログラミングレベルを手動で調整するには、まず、その動的チューニングを無効にする必要があります。アプリケーションの負荷をテストして、サーバースループットと要求の応答時間に対する、データベースサーバーのマルチプログラミングレベルの効果を分析することをおすすめします。Windows パフォーマンスモニターまたは UNIX 上のパフォーマンス統計ユーティリティ (dbstats) を使用すると、アプリケーションのテスト時にデータベースサーバーの動作を簡単に分析できます。PROPERTY、DB_PROPERTY、CONNECTION PROPERTY 関数を使用すると、パフォーマンス統計を問い合わせることができます。

  この分析中に、アクティブな要求やスケジュールされていない要求に関連のあるパフォーマンスカウンターを確認してください。 ActiveReq サーバープロパティ754 ページと UnschReq サーバープロパティ778 ページを参照してください。

  アクティブな要求の数が、-gn データベースサーバーオプションの値よりも常に小さい場合は、マルチプログラミングレベルを下げることを検討します。ただし、データベースサーバーの実行キューに対してタスクを追加するクエリ内並列処理の効果を考慮に入れる必要があります。クエリ内並列処理の効果が不十分である場合は、全体的なシステムスループットを低下させることなく、安全に、マルチプログラミングレベルを下げることができます。ただし、要求の総数 (アクティブな要求 + スケジュールされていない要求) が -gn で指定された値よりも大きくなることが多い場合は、マルチプログラミングレベルを上げることができます。マルチプログラミングレベルを上げた場合のパフォーマンスのトレードオフを検討してからレベルを変更してください。「マルチプログラミングレベルの増加」168 ページを参照してください。

**参照**

- 「-gn dbsrv12 サーバーオプション」237 ページ
- 「-gna dbsrv12 サーバーオプション」238 ページ
- 「-gnh dbsrv12 サーバーオプション」239 ページ
- 「-gnl dbsrv12 サーバーオプション」240 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』

## 各種マルチプログラミングレベル設定の効果

マルチプログラミングレベルの適切な値を決定するのは、難しいことがあります。たとえば、データベースアプリケーションが Java ストアドプロシージャーを利用する場合や、クエリ内並列処理が有効になっている場合、これらの要求を処理するために追加で作成されたサーバータスクは、現在のマルチプログラミングレベルを超えると、他の要求が完了するまで実行を待機することになります。このような状況では、マルチプログラミングレベルを上げることが適切です。ネットワークデータベースサーバーのマルチプログラミングレベルは、いつでも調整できます。ただし、パーソナルデータベースサーバーについては、サーバーの起動後にマルチプログラミングレベルを変更できません。

### マルチプログラミングレベルの増加

多くの場合、マルチプログラミングレベルを上げると、データベースサーバーの全体的なスループットも相応して向上します。これは、マルチプログラミングレベルを上げることによって同時に実行できるタスクの数が増えるためです。ただし、マルチプログラミングレベルを上げるときに検討が必要なトレードオフがあります。これには次のようなものがあります。

- **競合の増加**　同時に実行するタスクの数を増加すると、アクティブな要求間で競合が発生する可能性が高まることがあります。競合はリソースに関連するもので、スキーマロックやローロック、データベースサーバー内部のデータ構造や同期プリミティブなどが挙げられます。このような状況では、サーバーのスループットが実際に低下することがあります。

- **サーバーのオーバーヘッドの増加**　アクティブな各タスクは、ワーカーの割り付けと管理を要求します。スケジュールを制御するために追加の管理構造も必要です。さらに、各ワーカーは、実行スタックに対するアドレス領域の事前割り付けを要求します。スタックサイズはプラットフォームによって異なります。Windows オペレーティングシステムでは、スタック領域の割り付けはデータベースサーバープロセスのアドレス領域に影響を及ぼします。ただし、スタックメモリは要求に応じて割り付けられます。UNIX プラットフォームの場合 (Linux を含みます)、スタックの退避メモリが即座に割り付けられます。そのため、マルチプログラミングレベルを上げると、サーバーのメモリフットプリントが増加します。また、利用できるアドレス領域が少なくなるため、キャッシュマネージャーで利用できるメモリ量が減少します。

- **スラッシング**　データベースサーバーは、個々の要求に対する処理を実行するためではなく、実行のオーバーヘッドを管理するためだけに、多量のリソースを使用する状態に陥ることがあります。この状態は一般的に「スラッシング」と呼ばれています。スラッシングが発生するのは、データベースサーバーのキャッシュ内の領域に対して競合するアクティブなタスクの数があまりに多く、これらのタスクのセットが使用するデータベースページのワーキングセットを保てるほどキャッシュが大きくない場合などです。結果として、オペレーティングシステムで発生する場合と同様に、ページの横取りが発生することがあります。

- **クエリ処理への影響**　データベースサーバーは、同時に処理可能なメモリを大量に消費する要求の最大数を選択します。データベースサーバーのマルチプログラミングレベルを上げても、メモリが利用可能になるまで、要求が待機状態になる場合があります。「キャッシュとメモリガバナー」『SQL Anywhere サーバー SQL の使用法』を参照してください。

- **データ構造のメモリ**　データベースサーバーは、リソースを使用して文を解析したり、最適化したりします。複雑な文やキャッシュサイズが小さい場合、サーバーデータ構造に使用されるメモリのサイズが利用可能な上限を超えてしまう場合があります。各タスクのサーバーデータ構造に使用するメモリ量は、メモリガバナーによって制限されます。各タスクには、次の近似的な制限があります。

  (3/4 maximum-cache-size) / (number-of-active-requests)

  制限を超えると、文にはエラーが発生します。

### マルチプログラミングレベルの低下

同時に実行するタスクの数を減らしてデータベースサーバーのマルチプログラミングレベルを下げると、一般に、データベースサーバーのスループットは低下します。ただし、マルチプログ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

ラミングレベルを下げることで、個々の要求の応答時間は短縮されることがあります。これは、リソースに対して競合する要求の数が少なくなり、ロック競合の確率が低下するためです。

マルチプログラミングレベルを下げすぎると、スレッドデッドロックが発生することがあります。マルチプログラミングレベルが負荷に対して適切なレベルになっていて、スレッドデッドロックが発生する場合は、アプリケーション設計に問題があることが考えられます。結果として競合が発生し、延いてはスケーラビリティが低下します。例として、データベースに新しいデータを挿入するときにすべてのアプリケーションが変更を加える必要のあるテーブルについて考えてみます。この方法は、プライマリキーを生成するスキームの一部として、よく使用されます。このようなテーブルでは、結果として、アプリケーションの挿入トランザクションのすべてが事実上直列化される状況になります。共有テーブルが直列化されることに起因して、サーバーがサービスを提供する速度よりも挿入トランザクション速度の方が速くなると、通常、スレッドデッドロックが発生します。「デッドロック」『SQL Anywhere サーバー SQL の使用法』を参照してください。

-gnl データベースサーバーオプションは、最小マルチプログラミングレベルを設定します。

**参照**


- 「-gn dbsrv12 サーバーオプション」237 ページ
- 「-gna dbsrv12 サーバーオプション」238 ページ
- 「-gnh dbsrv12 サーバーオプション」239 ページ
- 「-gnl dbsrv12 サーバーオプション」240 ページ
- 「-gns dbsrv12 サーバーオプション」241 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「パフォーマンス統計値ユーティリティ (dbstats) (UNIX)」991 ページ
- 「Windows パフォーマンスモニターを使用した統計値のモニタリング」『SQL Anywhere サーバー SQL の使用法』
- 「接続、データベース、データベースサーバーのプロパティ」713 ページ


## マルチプログラミングレベルの手動調整

ネットワークサーバーのマルチプログラミングレベルを手動で調整できます。ただし、マルチプログラミングレベルの自動チューニングが有効になっている場合は、手動設定後にサーバーによって自動的にマルチプログラミングレベルが再調整されることがあります。マルチプログラミングレベルを下げた場合、サーバーは新しいマルチプログラミングレベルが有効になるまでアクティブなタスクの完了を待機する必要がある場合があるため、新しい設定は直ちに有効にならないことがあります。マルチプログラミングレベルを、MinMultiProgrammingLevel 設定と MaxMultiProgrammingLevel 設定によってそれぞれ指定された下限値と上限値を超える値に設定しようとすると、エラーが返されます。

マルチプログラミングレベルの自動チューニングは、データベースサーバーの起動時またはデータベースサーバーの実行中に無効にすることができます。

- **サーバー起動時の自動マルチプログラミングチューニングの無効化**　データベースサーバーを起動するときに、-gna 0 を指定して、マルチプログラミングレベルの自動チューニングを無効にします。

```
dbsrv12 -gna 0 ...
```

デフォルトのマルチプログラミングレベルを変更するには -gn オプションを使用し、最小値と最大値を設定するにはそれぞれ -gnl オプションと -gnh オプションを使用します。

- **実行中サーバーの自動マルチプログラミングチューニングの無効化**　データベースサーバーがすでに実行されている場合は、次の SQL 文を実行して、マルチプログラミングレベルの自動チューニングを無効にします。

```
CALL sa_server_option ( 'AutoMultiProgrammingLevel', 'NO' );
```

最小値と最大値を設定するには、それぞれ MinMultiProgrammingLevel プロパティと MaxMultiProgrammingLevel プロパティを使用します。次に例を示します。

```
CALL sa_server_option ( 'MinMultiProgrammingLevel', 10 );
CALL sa_server_option ( 'MaxMultiProgrammingLevel', 100 );
```

現在のマルチプログラミングレベルを設定するには、次のように CurrentMultiProgrammingLevel プロパティを使用します。

```
CALL sa_server_option ( 'CurrentMultiProgrammingLevel', 25 );
```

**参照**


- 「-gna dbsrv12 サーバーオプション」238 ページ
- 「-gn dbsrv12 サーバーオプション」237 ページ
- 「-gnh dbsrv12 サーバーオプション」239 ページ
- 「-gnl dbsrv12 サーバーオプション」240 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』


# 現在のセッション外でのデータベースサーバーの起動

ユーザー ID とパスワードを使用してコンピューターにログインすると、ユーザーは「セッション」を確立します。データベースサーバー、またはその他のアプリケーションを起動すると、そのセッション内で稼働します。コンピューターからログオフすると、そのセッションに関連するすべてのアプリケーションは停止します。

データベースサーバーを常時利用可能にする必要がある場合は、コンピューターからログオフしてもデータベースサーバーが稼働し続けるように、Windows と UNIX 用の SQL Anywhere を実行できます。

- **Windows サービス**　Windows データベースサーバーをサービスとして実行できます。この設定は、高可用性サーバーを実行する場合に便利です。「Windows サービス」171 ページを参照してください。

- **UNIX デーモン**　-ud オプションを使用すると、UNIX データベースサーバーをデーモンとして実行できます。これによって、データベースサーバーをバックグラウンドで実行し、ログオフ後も引き続き実行させることができます。「UNIX でのデーモンとしてのデータベースサーバー」180 ページを参照してください。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

- **Linux サービス**　Linux データベースサーバーをサービスとして実行できます。この設定には、高可用性サーバーを実行するための便利な特性が多数あります。「Linux 用サービスユーティリティ (dbsvc)」1007 ページを参照してください。

### サービスとして実行できるプログラム

サービスとして実行できるプログラムは次のとおりです。

- ネットワークデータベースサーバー (*dbsrv12*)
- パーソナルデータベースサーバー (*dbeng12*)
- SQL Remote Message Agent (dbremote)
- Mobile Link サーバー (*mlsrv12*)
- Mobile Link 同期クライアント (dbmlsync)
- Mobile Link Relay Server (*rshost*)
- Mobile Link Relay Server Outbound Enabler (*rsoe*)
- Mobile Link Listener ユーティリティ (*dblsn*) (Windows のみ)
- Mobile Link エージェント (*mlagent*) (Windows のみ)
- SQL Anywhere Broadcast Repeater ユーティリティ (*dbns12*)
- SQL Anywhere ボリュームシャドウコピーサービス (*dbvss12*) (Windows のみ)

これらのアプリケーションすべてが、SQL Anywhere のすべてのエディションに付属しているわけではありません。

### 参照


- 「Windows 用サービスユーティリティ (dbsvc)」1013 ページ
- 「Windows サービスの作成」172 ページ


## Windows サービス

データベースサーバーは、サービスではなく Microsoft Windows プログラムのように実行できます。ただし、標準のプログラムとして、また、特にマルチユーザー環境で実行する場合には制限があります。

### 標準実行プログラムとして実行する場合の制限事項

プログラムを起動すると、プログラムは Windows のログインセッションで実行されます。これは、コンピューターからログオフするとプログラムが停止されることを意味します。データベースサーバーで通常行われるように、プログラムを常に動作させておきたい場合、この設定によってコンピューターの用途が制限されます。データベースサーバーを実行し続けるには、そのデータベースサーバーを実行するコンピューターにログオンし続けます。この設定では、Windows コンピューターをログオンした状態にしておくことになるので、セキュリティ上の問題も発生します。

### サービスの利点

アプリケーションを Windows サービスとしてインストールすると、ログオフしても実行できます。

サービスを開始するときに、サービスは LocalSystem という特別なシステムアカウント (または ユーザーが指定した別のアカウント) を使ってログオンします。サービスを起動するユーザーの ID とサービスは関連していないので、起動した人がログオフしてもサービスは開いたままになります。Windows コンピューターが起動してユーザーがログオンする前に、サービスを自動的に開始するように設定することもできます。

### サービスの管理

Sybase Central は、Windows のサービス マネージャーよりも便利でわかりやすい方法で SQL Anywhere サービスを管理します。dbsvc ユーティリティを使用してサービスの作成や変更を行うこともできます。「Windows 用サービスユーティリティ (dbsvc)」1013 ページを参照してください。

## Windows サービスの作成

**♦ 新しいサービスを作成するには、次の手順に従います (Sybase Central の場合)。**

各ネットワークサーバーまたはパーソナルサーバーは、複数のデータベースを実行できます。複数のデータベースを同時に実行するときは、新しいサービスを作成するのではなく、既存のサービスに新しいデータベースを付加することをおすすめします。

1. 左ウィンドウ枠で、**[SQL Anywhere12]** をクリックします。
2. 右ウィンドウ枠で、**[サービス]** タブをクリックします。
3. **[ファイル]** » **[新規]** » **[サービス]** をクリックします。
4. サービス作成ウィザードの指示に従います。

**♦ 新しいサービスを作成するには、次の手順に従います (コマンドラインの場合)。**

各ネットワークサーバーまたはパーソナルサーバーは、複数のデータベースを実行できます。複数のデータベースを同時に実行するときは、新しいサービスを作成するのではなく、既存のサービスに新しいデータベースを付加することをおすすめします。

● -w オプションを指定してサービスユーティリティ (dbsvc) を実行します。

たとえば、データベースサーバーを LocalSystem ユーザーとして実行する myserv というパーソナルサーバーサービスを作成するには、次のコマンドを実行します。

```
dbsvc -as -w myserv "C:¥Program Files¥SQL Anywhere 12¥Bin32¥dbeng12.exe"
 -n william -c 8m "c:¥temp¥sample.db"
```

**注意**

●サービス名の最初の 8 文字は、ユニークにしてください。

●サービスを自動で開始するよう選択すると、コンピューターが Windows を起動するときに必ずサービスが開始されます。手動で開始することを選択した場合は、毎回 Sybase Central から開始する必要があります。今後使用するためにサービスを設定する場合は、**[無効]** をクリックします。

●Sybase Central でサービスを作成する場合は、実行プログラム自体の名前ではなく、実行プログラムのオプションをウィンドウで入力します。たとえば、20 MB のキャッシュサイズで myserver という名前のサンプルデータベースを指定して、ネットワークサーバーを実行する場合は、Sybase Central の**サービス作成ウィザード**の **[パラメーター]** テキストボックスに次のように入力します。

```
-c 20M
-n myserver "%SQLANY12%¥demo.db"
```

改行は任意です。

●サービスを実行するアカウントを選択します (LocalSystem という特別なアカウントまたは別のユーザー ID)。

この選択の詳細については、「アカウントオプション」175 ページを参照してください。

●Windows のデスクトップからサービスにアクセスできるようにする場合は、**[デスクトップとの対話をサービスに許可]** をクリックします。このチェックボックスをオフにすると、システムトレイにはアイコンが表示されず、デスクトップにもウィンドウが表示されません。

**参照**


- 「Windows 用サービスユーティリティ (dbsvc)」1013 ページ
- 「Windows サービスの設定」174 ページ
- 「現在のセッション外での Mobile Link サーバーの実行」『Mobile Link サーバー管理』


## Windows サービスの削除

**♦ Windows サービスを削除するには、次の手順に従います (Sybase Central の場合)。**

サービスを削除すると、サービスのリストからサーバー名が削除されます。サービスを削除しても、ハードディスクからソフトウェアが削除されることはありません。

前に削除したサービスを再インストールするには、オプションを再入力する必要があります。実行中のサービスは削除できません。

1. 左ウィンドウ枠で、**[SQL Anywhere12]** をクリックします。

   右ウィンドウ枠で、**[サービス]** タブをクリックします。

2. 右ウィンドウ枠で、削除するサービスを選択し、**[編集]** » **[削除]** をクリックします。

**♦ Windows サービスを削除するには、次の手順に従います (コマンドラインの場合)。**

サービスを削除すると、サービスのリストからサーバー名が削除されます。サービスを削除しても、ハードディスクからソフトウェアが削除されることはありません。

前に削除したサービスを再インストールするには、オプションを再入力する必要があります。実行中のサービスは削除できません。

- -d オプションを指定してサービスユーティリティ (dbsvc) を実行します。

  たとえば、myserv というサービスを確認プロンプトを表示させずに削除するには、次のコマンドを実行します。

  ```
  dbsvc -y -d myserv
  ```

**参照**

- 「Windows 用サービスユーティリティ (dbsvc)」1013 ページ

## Windows サービスの設定

サービスは、一連のオプションを使ってデータベースサーバーやその他のアプリケーションを実行します。

サービスは、オプションに加えて、サービスが実行されるアカウントを指定するパラメーターと、起動時の条件を指定するパラメーターを許可します。

**♦ サービスのパラメーターを変更するには、次の手順に従います (Sybase Central の場合)。**

1. 左ウィンドウ枠で、**[SQL Anywhere12]** をクリックします。
2. 右ウィンドウ枠で、変更するサービスを選択します。
3. **[ファイル]** » **[プロパティ]** をクリックします。
4. **[サービスのプロパティ]** ウィンドウのタブで、必要に応じてパラメーターを変更します。
5. 変更が完了したら **[OK]** をクリックします。

サービス設定の変更は、次回のサービス実行時から有効となります。**起動**オプションは、次回 Windows を起動するときに適用されます。

### Windows サービス起動オプション

次のオプションは、SQL Anywhere サービスの起動時の動作を制御します。これらのオプションは、**[サービスのプロパティ]** ウィンドウの **[一般]** タブで設定できます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

- **[自動]**　サービスは、Windows オペレーティングシステムが起動すると必ず起動します。この設定は、データベースサーバーやその他の常に稼働しているアプリケーションに適しています。
- **[手動]**　サービスは Administrator 権限を持つユーザーが起動したときにのみ起動します。Administrator 権限については、Windows のマニュアルを参照してください。
- **[無効]**　サービスは起動しません。

## Windows サービスオプション

サービスのオプションは、実行プログラムのものと同じです。

> **警告**
> **[サービスのプロパティ]** ウィンドウの **[設定]** タブには、サービスのオプションを指定する **[パラメーター]** テキストボックスがあります。このボックスには、実行プログラムの名前は入力しないでください。

### 例

2 つのデータベースを実行する my_server というネットワークサーバーサービスを、キャッシュサイズ 20 MB で起動するには、**[パラメーター]** フィールドに次のように入力します。

```
-c 20M
-n my_server
c:¥db_1.db
c:¥db_2.db
```

サンプルデータベースにユーザー ID DBA で接続する SQL Remote Message Agent サービスを起動するには、次のように入力します。

```
-c "UID=DBA;PWD=sql;DBN=demo"
```

## アカウントオプション

サービスが実行されるアカウントを選択できます。デフォルトでは、ほとんどのサービスは LocalSystem という特別なアカウントの下で実行されます。**[サービスのプロパティ]** ウィンドウの **[アカウント]** タブを開き、アカウント情報を入力すると、別のアカウントでログオンするようにサービスを設定できます。

LocalSystem 以外のアカウントでサービスを実行するには、そのアカウントに**サービスとしてログオン**する権限が必要です。この権限は、Windows のユーザー マネージャーの [高度なユーザー権利] で付与できます。この権限を付与するためにサービスユーティリティ (dbsvc) を使用することもできます。

### システムトレイにアイコンが表示される場合

- サービスが LocalSystem で実行されているときに、**[サービスのプロパティ]** ウィンドウの **[デスクトップとの対話をサービスに許可]** がオンになっている場合は、サービスを実行するコン

ピューター上の Windows にどのユーザーがログインしてもデスクトップにアイコンが表示されます。すべてのユーザーがアプリケーションウィンドウを開き、サービスとして実行されているプログラムを停止できます。

- サービスが LocalSystem で実行されているときに、**[サービスのプロパティ]** ウィンドウの **[デスクトップとの対話をサービスに許可]** がオフになっている場合は、ユーザーのデスクトップにアイコンは表示されません。サービスの状態を変更する権限を与えられているユーザーのみ、サービスを停止できます。

- サービスが他のアカウントで実行されている場合、デスクトップにアイコンは表示されません。サービスの状態を変更する権限を与えられているユーザーのみ、サービスを停止できます。

## 実行ファイルの変更

サービスに関連付けられたプログラム実行ファイルを変更するには、**[サービスのプロパティ]** ウィンドウの **[設定]** タブをクリックし、[ファイル名] テキストボックスに新しいパスとファイル名を入力します。

実行ファイルを新しいディレクトリに移動する場合は、この内容も変更します。

## サービスへの新しいデータベースの追加

**♦ 新しいデータベースを既存のサービスに追加するには、次の手順に従います (Sybase Central の場合)。**

各ネットワークサーバーまたはパーソナルサーバーは、複数のデータベースを実行できます。複数のデータベースを同時に実行するときは、新しいサービスを作成するのではなく、既存のサービスに新しいデータベースを付加することをおすすめします。

1. **[コンテキスト]** ドロップダウンリストから **[SQL Anywhere 12]** をクリックします。
2. 右ウィンドウ枠で、**[サービス]** タブをクリックします。
3. サービスを選択し、**[ファイル]** » **[プロパティ]** をクリックします。
4. **[設定]** タブをクリックします。
5. 新しいデータベースのパスとファイル名を、**[パラメーター]** ボックスのオプションリストの最後に追加します。
6. **[OK]** をクリックして変更を保存します。

   次回サービスを開始したときに、新しいデータベースが起動します。

稼働中のサーバー上でデータベースを起動するには、Interactive SQL などのクライアントアプリケーションを使います。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Interactive SQL からデータベースを起動する場合の詳細については、「START DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

Embedded SQL アプリケーションでこの機能を実装する方法については、「db_start_database 関数」『SQL Anywhere サーバー プログラミング』を参照してください。

アプリケーションからデータベースを起動しても、データベースはサービスに付加されません。サービスを停止して再起動すると、追加のデータベースは自動的には起動しません。

## 再表示頻度の設定

**♦ 再表示頻度を設定するには、次の手順に従います (Sybase Central の場合)。**

デフォルトでは、Sybase Central は各 Windows サービスのステータス (開始または停止) をチェックし、**[ステータス]** カラムを更新して現在のステータスを表示します。さらに、接続やロックなどの動的オブジェクト、イベントやメンテナンスプランの動的プロパティを再表示します。デフォルトでは、10 秒ごとに再表示します。自動再表示をオフにした場合、ステータスの変更を確認するには **[表示]** » **[フォルダーの再表示]** をクリックします。

1. **[ツール]** » **[SQL Anywhere 12]** » **[ユーザー設定]** をクリックします。
2. **[自動的に再表示]** タブをクリックします。
3. **[自動再表示を有効にする]** をクリックします。
4. **[再表示間隔 *X* 秒]** に時間間隔を指定します。デフォルトは 10 秒です。

   このウィンドウで設定した値は、変更するまでそのまま使われます。
5. **[適用]** をクリックします。
6. **[OK]** をクリックします。

## サービスの開始と停止

**♦ サービスを開始または停止するには、次の手順に従います (Sybase Central の場合)。**

サービスを開始すると、停止するまで実行を続けます。Sybase Central を閉じたり、ログオフしてもサービスは停止しません。

データベースサーバーのサービスを停止すると、データベースへの接続はすべて閉じられ、データベースサーバーは停止されます。他のアプリケーションのプログラムは終了します。

1. **[コンテキスト]** ドロップダウンリストから **[SQL Anywhere 12]** をクリックします。
2. 右ウィンドウ枠で、**[サービス]** タブをクリックします。
3. サービスを選択し、**[ファイル]** » **[スタート]** または **[停止]** を選択します。

## Windows サービスマネージャー

SQL Anywhere のすべてのサービス管理は、Sybase Central から行うことができます。Windows の **[コントロール パネル]** にある**サービス マネージャー**を使用していくつかのタスクを実行することはできますが、Windows の**サービス マネージャー**で SQL Anywhere サービスをインストールしたり設定したりすることはできません。

Windows の **[コントロール パネル]** から **サービス マネージャー**を開くと、サービスのリストが表示されます。SQL Anywhere サービスの名前は、サービスをインストールしたときに入力したサービス名の前に "SQL Anywhere" が付いた形式になっています。インストールされたすべてのサービスが、リストに表示されます。

## サービスグループの概要

システムの各サービスを、サービスグループの各メンバーに割り当てることができます。デフォルトでは、次の表に示すように各サービスがグループに属しています。

| サービス | デフォルトのグループ |
|---|---|
| ネットワークサーバー | SQLANYServer |
| パーソナルサーバー | SQLANYEngine |
| Mobile Link 同期クライアント | SQLANYMLSync |
| SQL Remote Message Agent | SQLANYRemote |
| Mobile Link サーバー | SQLANYMobiLink |
| SQL Anywhere Broadcast Repeater ユーティリティ | SQLANYNS |
| Mobile Link Listener | SQLANYLSN |
| SQL Anywhere ボリュームシャドウコピーサービス | SQLANYVSS |
| Mobile Link Relay Server | SQLANYRSHOST |
| Mobile Link Relay Server Outbound Enabler | SQLANYRSOE |
| Mobile Link エージェント | SQLANYMLAGENT |

サービスが適切なグループのメンバーであることをチェックしてから、サービスが正しい順序で開始するように設定してください。Sybase Central では、サービスが割り当てられているグループを調べ、このグループを変更できます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**♦ サービスが割り当てられているグループを調べて変更するには、次の手順に従います (Sybase Central の場合)。**

1. 左ウィンドウ枠で、**[SQL Anywhere12]** をクリックします。
2. 右ウィンドウ枠で、**[サービス]** タブをクリックします。
3. サービスを選択し、**[ファイル]** » **[プロパティ]** をクリックします。
4. **[依存性]** タブをクリックします。最上部のテキストボックスにサービスが割り当てられているグループの名前が表示されます。
5. **[変更]** をクリックして、システムで使用可能なグループのリストを表示します。
6. いずれかのグループを選択するか、新しいグループの名前を入力します。
7. **[OK]** をクリックして、そのグループにサービスを割り当てます。

## サービス依存性の管理

状況によっては、複数の実行プログラムをサービスとして実行する必要があり、これらの実行プログラムが相互に依存する場合があります。たとえば、レプリケーションを補助するために、サーバーと SQL Remote Message Agent を実行する場合があります。サービスは正しい順序で開始する必要があります。サーバーが起動する前に SQL Remote Message Agent サービスが開始されると、サーバーを検出できないためエラーになります。「サービスグループ」を使用すると、これらの問題を防ぐことができます。サービスグループは Sybase Central で管理します。Sybase Central を使用して、サービスの「依存性」を指定できます。次に例を示します。

- サービスグループのリストにある各グループの少なくとも 1 つのメンバーを開始してから現在のサービスを開始するようにできます。
- 任意の数のサービスを開始してから現在のサービスを開始するようにできます。たとえば、特定のネットワークサーバーを起動してから、そのサーバーに対して実行する SQL Remote Message Agent を開始するようにしたい場合などです。

**♦ サービスまたはグループを依存性のリストに追加するには、次の手順に従います (Sybase Central の場合)。**

1. **[コンテキスト]** リストから **[SQL Anywhere 12]** をクリックします。
2. 右ウィンドウ枠で、**[サービス]** タブをクリックします。
3. サービスを選択し、**[ファイル]** » **[プロパティ]** をクリックします。
4. **[依存性]** タブをクリックします。
5. **[サービスの追加]** または **[サービスグループの追加]** をクリックして、サービスまたはグループを依存性リストに追加します。
6. リストからサービスまたはグループを 1 つ選択します。

7. **[OK]** をクリックして、そのサービスまたはグループを依存性リストに追加します。

## UNIX でのデーモンとしてのデータベースサーバー

UNIX データベースサーバーをバックグラウンドで実行し、現在のセッションから独立して稼働させるには、データベースサーバーを「デーモン」として実行します。

> **注意**
> データベースサーバーをバックグラウンドで実行するのに '&' を使用しない。データベースサーバーをバックグラウンドで実行するのに UNIX の & (アンパサンド) コマンドを使用しても機能しません。サーバーがただちに停止するか応答しなくなります。データベースサーバーはデーモンとして実行してください。
>
> 同様に、一般的な fork()-exec() シーケンスを使用してプログラムの中からサーバーをバックグラウンドで起動しようとしても機能しません。これを行うには、データベースサーバーオプションのリストに -ud オプションを追加します。

UNIX データベースサーバーは、次のいずれかの方法でデーモンとして実行できます。

1. データベースサーバーの起動時に、-ud オプションを使用する。次に例を示します。

   ```
   dbsrv12 -ud demo
   ```

2. dbspawn ツールを使用してデータベースサーバーを起動する。次に例を示します。

   ```
   dbspawn dbsrv12 demo
   ```

   dbspawn を使用することの利点は、デーモンが起動し、要求を受け入れる状態になったことを確認するまで dbspawn プロセスが停止しないことです。何らかの理由でデーモンの起動が失敗した場合、dbspawn の終了コードは 0 以外の値になります。

   -ud オプションを使用してデーモンを直接起動したときは、dbeng12 コマンドと dbsrv12 コマンドがデーモンプロセスを作成し、(終了して次のコマンドを実行できるように) すぐに返します。その後、デーモンがそれ自体を初期化するか、コマンドで指定されたデータベースを開こうとします。

   データベースサーバーを使用できる 1 つ以上のアプリケーションでデーモンが確実に実行されるようにしたい場合、アプリケーションの起動前にデーモンを実行する dbspawn を使用します。次の例では、csh スクリプトを使用してこれをテストする方法を示します。

   ```
   #!/bin/csh
   # start the server as a daemon and ensure that it is
   # running before you start any applications
   dbspawn dbsrv12 demo
   if ( $status != 0 ) then
     echo Failed to start demo server
     exit
   endif
   # ok, now you can start the applications
   ...
   ```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

次の例では、sh スクリプトを使用して、アプリケーションの起動前にデーモンが実行中であるかどうかをテストします。

```
#!/bin/sh
# start the server as a daemon and ensure that it is
# running before you start any applications
dbspawn dbsrv12 demo
if [ $? != 0 ]; then
   echo Failed to start demo server
   exit
fi
# ok, now you can start the applications
...
```

3. C プログラムの中からデーモンを生成する。次に例を示します。

```
...
if( fork() == 0 ) {
      /* child process = start server daemon */
      execl( "/opt/sqlanywhere12/bin/dbsrv12",
"dbsrv12", "-ud", "demo" );
   exit(1);
}
/* parent process */
...
```

-ud オプションが使用されていることに注意してください。

**参照**

- 「-ud dbeng12/dbsrv12 サーバーオプション」282 ページ
- 「サーバーバックグラウンド起動ユーティリティ (dbspawn)」1022 ページ

# 認証 SQL Anywhere アプリケーション

SQL Anywhere の OEM Edition は、アイエニウェア OEM パートナーに提供されています。SQL Anywhere OEM Edition を使用すると、認証アプリケーションは、ユーザー ID に付与されたパーミッションに基づいて、データベース上で任意の操作を実行できます。

非認証で接続すると、読み込み専用アクセスとなり、テンポラリテーブルに対する挿入、更新、削除を実行できます。非認証接続を使用すると、ストアドプロシージャーを使用して複雑なレポートを作成したり、Crystal Reports などのレポーティングツールを使用してアクセスしたりすることができます。

認証メカニズムは、他のすべてのアプリケーションプログラミング言語やツールとは無関係であり、接続ごとに独立して実行されます。したがって、アプリケーション内で、認証接続と、制限付きの非認証接続の両方を使用できます。

認証は、セキュリティメカニズムではありません。データベースに対して非認証データベースサーバーを実行するユーザーは、標準的な SQL パーミッションスキームに基づいて任意の操作を実行できます。

# 認証アプリケーションの開発

認証アプリケーションの開発は、単純なプロセスです。特殊な認証シグネチャーがデータベースに組み込まれ、もう 1 つのシグネチャーがアプリケーションに組み込まれます。アプリケーションがデータベースに接続すると、シグネチャーが比較され、アプリケーションの認証が行われます。SQL Anywhere の認証アプリケーションを開発するには、以下の手順が必要です。

1. 「認証シグネチャーの取得」182 ページ
2. 「データベースの認証」183 ページ
3. 「アプリケーションの認証」184 ページ

SQL Anywhere に付属しているデータベースツール (Sybase Central、Interactive SQL、および dbbackup などのユーティリティ) は、自己認証形式です。これらのツールは、認証データベースへの操作が制限されません。データベース自体が認証されていない場合は、ツールは、制限された読み込み専用モードで動作します。

認証アプリケーションでは、OEM Edition の SQL Anywhere データベースサーバーを使用する必要があります。このエディションで、通常のデータベースサーバーと唯一異なる点は、認証命令を処理することです。認証命令は、その他のエディションのデータベースサーバーでは無視されます。認証データベースサーバーを使用しない場合は、非認証アプリケーションにも制限は適用されません。

## 認証シグネチャーの取得

♦ **認証シグネチャーを取得するには、次の手順に従います。**

> **注意**
> 認証シグネチャーを取得するには、アイエニウェアとの OEM 契約が必要です。

1. http://www.sybase.com/sql_anywhere_authentication_registration に移動してください。
2. フォームに記入し、認証シグネチャーを取得します。認証メカニズムに次の情報が組み込まれます。
   - **会社名**　会社の名前。
   - **アプリケーション名**　アプリケーションの名前。

   フォームを完成させると、48 時間以内に、データベースシグネチャーとアプリケーションシグネチャーについて知らせる電子メールが送信されます。これらのシグネチャーは、英字と数字で構成される長い文字列です。認証情報を含む電子メールメッセージには、情報の使用法についていくつかの例が記載されています。電子メールシステムの中には、命令の途中で強制的に改行するものがあります。命令を正常に機能させるように、電子メールメッセージに挿入された改行を削除して行を再結合してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

会社名やアプリケーション名が認証メカニズムに組み込まれる方法の詳細については、「データベースの認証」183 ページを参照してください。

## データベースの認証

Authenticated データベースプロパティを使用して、データベースが認証されたかどうかを判別できます。

```
SELECT DB_PROPERTY ( 'Authenticated' );
```

データベースプロパティの詳細については、「データベースプロパティ値のアクセス」781 ページを参照してください。

**♦ データベースを認証するには、次の手順に従います。**

1. 次の SQL 認証文を使用して、データベースの database_authentication オプションを設定します。

```
SET OPTION PUBLIC.database_authentication='company=company-name;
    application=application-name;
    signature=database-signature';
```

> **注意**
> 構文例には、読みやすいように改行が追加されています。ただし構文を実行する場合には、改行なし、および等号とセミコロンの間のスペースなしで実行してください。

2. *company-name* と *application-name* の引数は、シグネチャーの取得時にアイエニウェアに提供した値です。*database-signature* は、アイエニウェアから受け取ったデータベースシグネチャーです。

3. データベースを再起動して、設定を有効にします。

データベースサーバーが認証データベースをロードするとき、データベースサーバーメッセージウィンドウに、認証された会社とアプリケーションについて説明するメッセージが表示されます。このメッセージを調べて、database_authentication オプションが有効になっているかどうかを確認できます。メッセージの形式は次のとおりです。

```
このデータベースは次の使用を目的としてライセンスされています。
アプリケーション：application-name
会社：company-name
```

> **ヒント**
> SQL スクリプトファイルに認証文を格納し、長いシグネチャーを繰り返し入力することを回避できます。**[ファイル]** » **[スクリプトの実行]** をクリックして、Interactive SQL から SQL スクリプトファイルを実行できます。
>
> SQL Anywhere インストールディレクトリの *scripts* サブフォルダーに、ファイル *authenticate.sql* を作成し、このファイルに認証文を格納すると、この文は、データベースの作成、再構築、またはアップグレードを行うたびに適用されます。「認証データベースのアップグレード」186 ページを参照してください。

## アプリケーションの認証

認証アプリケーションは、接続確立後すぐに、connection_authentication データベースオプションを設定する必要があります。このオプションは、接続が確立された直後にすべての接続に対して行われなければなりません。ODBC アプリケーションや JDBC アプリケーションは、データベースに対して機能について問い合わせます。ユーザーは、これらのアクションについて制御する必要はありません。このため、すべての接続には、制限が適用される前に 30 秒間の猶予期間が設けられています。猶予期間があることにより、使用している開発ツールには関係なく、アプリケーションは認証を行うことができます。

Authenticated 接続プロパティを使用して、現在の接続が認証されたかどうかを判別できます。

```
SELECT CONNECTION_PROPERTY ( 'Authenticated' );
```

接続プロパティの詳細については、「接続プロパティ値のアクセス」713 ページを参照してください。

次の SQL 文は、接続を認証します。

```
SET TEMPORARY OPTION connection_authentication='company = company-name;
    application=application-name;
    signature=application-signature';
```

**注意**
構文例には、読みやすいように改行が追加されています。ただし構文を実行する場合には、改行なし、および等号とセミコロンの間のスペースなしで実行してください。

このオプションは、TEMPORARY キーワードを使用して、接続の間だけ設定できます。*company-name* と *application-name* は、データベース認証の文と一致する必要があります。*application-signature* は、Sybase から取得するシグネチャーです。

データベースサーバーは、データベースシグネチャーに対してアプリケーションシグネチャーを検証します。シグネチャーが検証されると、接続が認証され、SQL パーミッションで設定されている内容にかかわらず、アクティビティは制限されません。シグネチャーが検証されないと、接続は、非認証アプリケーションによって許可されているアクションに制限されます。

## 認証文の実行

認証オプションを設定する SET TEMPORARY OPTION 文を実行する方法は、使用するプログラミングインターフェイスによって異なります。ここに示すシグネチャーは、有効なシグネチャーではありません。次のインターフェイスを使用して認証オプションを設定することを前提に、例を示します。

- ODBC
- Sybase PowerBuilder
- JDBC
- ADO.NET
- Embedded SQL

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**注意**
会社名に引用符やアポストロフィなどの特殊文字が含まれている場合 (たとえば、Joe's Garage など)、認証文の構成には注意が必要です。認証オプションセット全体 (Company=...;Application=...;Signature=...) で SQL 文字列になります。SQL における文字列の規則では、文字列の中に引用符を含める場合は、その文字を 2 つ続けて記述することになっています。次に例を示します。

```
SET TEMPORARY OPTION connection_authentication='Company = Joe''s Garage;
   Application=Joe''s Program;
   Signature=0fa55157edb8e14d818e...';
```

構文例には、読みやすいように改行が追加されています。ただし構文を実行する場合には、改行なし、および等号とセミコロンの間のスペースなしで実行してください。

### ODBC

次の文を使用します。

```
SQLExecDirect(
   hstmt,
   "SET TEMPORARY OPTION connection_authentication=
     'Company=MyCo;
      Application=MyApp;
      Signature=0fa55159999e14d818e...';",
   SQL_NTS
);
```

文字列は、1 行に入力します。または、連結して構築します。

### Sybase PowerBuilder

次の PowerScript 文を使用します。

```
EXECUTE IMMEDIATE
  "SET TEMPORARY OPTION connection_authentication=
     'Company=MyCo;
      Application=MyApp;
      Signature=0fa551599998e14d818e...';"
USING SQLCA
```

### [JDBC]

次の文を使用します。

```
Statement Stmt1 = con.createStatement();
Stmt1.executeUpdate(
   "SET TEMPORARY OPTION connection_authentication=
   'Company=MyCo;
    Application=MyApp;
    Signature=0fa55159999e14d818e...';"
);
```

文字列は、1 行に入力します。または、連結して構築します。

### [ADO.NET]

次の文を使用します。

```
SACommand cmd=new SACommand(
  "SET TEMPORARY OPTION connection_authentication=
     'Company=MyCo;
      Application=MyApp;
      Signature=0fa551599998e14d818e...';",
   con
);
cmd.ExecuteNonQuery();
```

文字列は、1 行に入力します。または、連結して構築します。

**[Embedded SQL]**

```
Use the following statement:
EXEC SQL SET TEMPORARY OPTION connection_authentication=
  'Company=MyCo;
   Application=MyApp;
   Signature=0fa551599998e14d818e...';
```

文字列は、1 行に入力します。または、連結して構築します。

認証データベースに接続するとき、接続と認証の手順は別々に行います。ただし、Microsoft Visual Basic Grid オブジェクトなどの一部のオブジェクトは、独立した暗黙的な接続の試行が可能であり、自動的に認証が行われません。このような場合は、接続は認証されず、データベース操作は失敗します。この問題を回避するには、接続文字列に InitString 接続パラメーターを指定します。次の例は、すべての接続で接続確立直後に認証が行われるように、InitString 接続パラメーターを含めるように Microsoft Visual Basic アプリケーションを修正する例を示したものです。

```
mConnectionString =
  "Provider=SAPROV.12;
   UID=DBA;
   PWD=sql;
   Host=test12;
   InitString=SET TEMPORARY OPTION connection_authentication=
     'Company=MyCo;
      Application=MyApp;
      Signature=0fa55157edb8e14d818e...'"
mdbName.ConnectionString = mConnectionString
mdbName.Open
mIsSQL = True
```

## 認証データベースのアップグレード

**♦ 認証データベースをアップグレードまたは再構築するには、次の手順に従います。**

データベースのアップグレードや再構築を行ったときに認証情報を保存するには、ファイル *authenticate.sql* に認証文を格納するしか方法がありません。

CREATE DATABASE 文と ALTER DATABASE UPGRADE 文、dbupgrad ユーティリティと dbunload ユーティリティ、**データベースアップグレードウィザード**は、*%SQLANY12%/scripts* ディレクトリに *authenticate.sql* という名前のファイルが存在するかどうかを確認し、存在する場合はその内容を実行します。このファイルを作成してからデータベースのアップグレードまたは再構築を実行してください。

1. *authenticate.sql* というファイルを *%SQLANY12%¥scripts* ディレクトリに作成します。
2. 次の内容をファイルに追加します。

   ```
   SET OPTION PUBLIC.database_authentication = 'authentication-statement'
   go
   ```

   ファイルには go が含まれている必要があります。含まれていないと、この文は無視されます。

   *authentication-statement* 文字列の内容については、「database_authentication」612 ページを参照してください。
3. データベースをアップグレードまたは再構築します。手順は、アップグレードまたは再構築するデータベースファイルのバージョンによって異なります。次の項を参照してください。
   - 「バージョン 9 以前のデータベースのバージョン 12 用への再構築」『SQL Anywhere 12 変更点とアップグレード』
   - 「バージョン 10 以降のデータベースのアップグレード手順」『SQL Anywhere 12 変更点とアップグレード』
   - 「バージョン 10 以降のデータベースの再構築手順」『SQL Anywhere 12 変更点とアップグレード』

# Windows Vista、Windows 7、Windows Server 2008、Windows Server 2008 R2 での SQL Anywhere

SQL Anywhere ソフトウェアを Vista、Windows 7、Windows Server 2008、Windows Server 2008 R2 で実行する場合は、次の考慮事項が当てはまります。

- **セキュリティ**　これらのオペレーティングシステムには、ユーザーアカウント制御 (UAC) という新しいセキュリティモデルが採用されています。UAC はデフォルトで有効に設定され、ファイルに書き込み可能とされるプログラムの動作に影響する可能性があります (特に、コンピューターが複数のユーザーをサポートしている場合)。ファイルとディレクトリを作成した場所と作成方法によって、あるユーザーが作成したファイルを、他のユーザーが読み込んだり、書き込むことが許可されなくなる場合があります。SQL Anywhere をデフォルトのディレクトリにインストールした場合は、複数のユーザーに読み込み／書き込みアクセスを許可する必要があるファイルおよびディレクトリが適切に設定されます。

- **SQL Anywhere 昇格操作エージェント**　UAC がアクティブな状態で実行する場合に、特定のアクションで権限の昇格が必要になります。SQL Anywhere では、次のプログラムで昇格が必要になることがあります。

  - *dbdsn.exe*
  - *dbelevate12.exe*
  - *dblic.exe*
  - *dbsvc.exe*
  - *installULNet.exe*
  - *mlasinst.exe*
  - *SetupVSPackage.exe*
  - *ulcond12.exe*

  次の DLL では、登録または登録の解除時に昇格が必要です。

  - *dbctrs12.dll*
  - *dbodbc12.dll*
  - *dboledb12.dll*
  - *dboledba12.dll*

  UAC がアクティブな場合、SQL Anywhere の昇格操作エージェントに対して昇格を確認するメッセージが表示されることがあります。このメッセージは、識別されたプログラムの実行を継続するかどうかを確認したり (管理者としてログオンしている場合)、管理者のクレデンシャルを提供するように求める (管理者以外でログオンしている場合) ため、ユーザーアカウント制御システムによって発行されるものです。

- **配備に関する考慮事項**　プログラム *dbelevate12.exe* は、昇格された権限が必要な操作を実行するために SQL Anywhere コンポーネントによって内部的に使用されます。この実行プログラムは、SQL Anywhere の配備環境に含まれている必要があります。

- **Microsoft ActiveSync のサポート**　Microsoft ActiveSync ユーティリティは、Vista、Windows 7、Windows 2008 ではサポートされていません。Windows Mobile Device Center で置き換えられました。SQL Anywhere ActiveSync プロバイダーインストールユーティリティを Windows Mobile Device Center で使用できます。

- **署名された SQL Anywhere 実行プログラム**　SQL Anywhere 実行プログラムは、iAnywhere Solutions, Inc. によって署名されています。

- **Windows サービス**　Windows Vista、Windows 7、Windows 2008 に対応しているサービスは、デスクトップと対話できません。これらのオペレーティングシステムでは、(サービス定義で **[デスクトップとの対話をサービスに許可]** が有効になっている場合でも) SQL Anywhere サービスはデスクトップと対話しません。SQL Anywhere データベースサーバーは、Sybase Central または SQL Anywhere コンソールユーティリティ (dbconsole) からモニタリングできます。

  Sybase Central をこれらのオペレーティングシステムで実行している場合、サービスがデスクトップと対話できるオプションは無効になります。

- **AWE キャッシュの使用**　AWE キャッシュを使用するには、データベースサーバーを管理者として実行する必要があります。AWE キャッシュを使用して昇格していないデータベース

サーバーを起動すると、AWE を使用できるように管理者としてデータベースサーバーを実行するように要求する警告が表示されます。「-cw dbeng12/dbsrv12 サーバーオプション (旧式)」217 ページを参照してください。

**注意**
AWE は使用されなくなりました。サイズの大きいキャッシュが必要な場合は、64 ビット版の Windows オペレーティングシステムで 64 ビット版の SQL Anywhere データベースサーバーを使用することをおすすめします。

# SQL Anywhere データベースサーバーの構文

パーソナルデータベースサーバーまたはネットワークデータベースサーバーを起動します。

### 構文

{ **dbeng12** | **dbsrv12** }
[ *server-options* ] [ *database-file* [ *database-options* ] ...]

### サーバーオプション

| サーバーオプション | 説明 |
|---|---|
| *@data* | 設定ファイルまたは環境変数からオプションを読み込みます。「@data dbeng12/dbsrv12 サーバーオプション」201 ページを参照してください。 |
| **-?** | 使用法を表示します。「-? dbeng12/dbsrv12 サーバーオプション」203 ページを参照してください。 |
| **-b** | バルクオペレーションモードで実行します。「-b dbeng12/dbsrv12 サーバーオプション」203 ページを参照してください。 |
| **-c** *size* | 初期キャッシュサイズを設定します。 「-c dbeng12/dbsrv12 サーバーオプション」204 ページを参照してください。 |
| **-ca 0** | 動的なキャッシュサイズの設定を無効にします (Windows、UNIX、Mac OS X)。「-ca dbeng12/dbsrv12 サーバーオプション」206 ページを参照してください。 |
| **-cc**{ + \| - } | キャッシュウォーミングに使用するデータベースページに関する情報を収集します。「-cc dbeng12/dbsrv12 サーバーオプション」207 ページを参照してください。 |
| **-ch** *size* | キャッシュサイズの上限値を設定します (Windows、UNIX、Mac OS X)。「-ch dbeng12/dbsrv12 サーバーオプション」208 ページを参照してください。 |

| サーバーオプション | 説明 |
|---|---|
| **-chx** *size* | キャッシュ以外に使用するアドレス領域を確保します (32 ビット Windows、32 ビット UNIX)。「-ch dbeng12/dbsrv12 サーバーオプション」208 ページを参照してください。 |
| **-cl** *size* | キャッシュサイズの下限値を設定します (Windows、UNIX、Mac OS X)。「-cl dbeng12/dbsrv12 サーバーオプション」211 ページを参照してください。 |
| **-cm** *size* | Address Windowing Extensions (AWE) キャッシュに割り付けるアドレス領域のサイズを指定します (Windows)。「-cm dbeng12/dbsrv12 サーバーオプション」212 ページを参照してください。 |
| **-cp** *location*[ **;***location* ... ] | クラスを検索する一連のディレクトリまたは JAR ファイルを指定します。「-cp dbeng12/dbsrv12 サーバーオプション」213 ページを参照してください。 |
| **-cr**{ + \| - } | データベースページを保持するキャッシュを準備します。「-cr dbeng12/dbsrv12 サーバーオプション」214 ページを参照してください。 |
| **-cs** | データベースサーバーメッセージウィンドウにキャッシュの使用状況を表示します。「-cs dbeng12/dbsrv12 サーバーオプション」215 ページを参照してください。 |
| **-cv**{ + \| - } | データベースサーバーメッセージウィンドウでのキャッシュウォーミングに関するメッセージの表示を制御します。「-cv dbeng12/dbsrv12 サーバーオプション」216 ページを参照してください。 |
| **-cw** | データベースサーバーキャッシュサイズの設定を目的とした Address Windowing Extensions の使用を有効にします (Windows)。「-cw dbeng12/dbsrv12 サーバーオプション (旧式)」217 ページを参照してください。 |
| **-dt** *temp-file-dir* | テンポラリファイルを保存するディレクトリを指定します。「-dt dbeng12/dbsrv12 サーバーオプション」221 ページを参照してください。 |
| **-ec** *encryption-options* | パケットの暗号化を有効にします (ネットワークサーバー)。「-ec dbeng12/dbsrv12 サーバーオプション」223 ページを参照してください。 |
| **-ep** | 暗号化キーを入力するよう要求します。「-ep dbeng12/dbsrv12 サーバーオプション」226 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| サーバーオプション | 説明 |
|---|---|
| **-es** | 共有メモリを経由した暗号化されていない接続を許可します。「-es dbeng12/dbsrv12 サーバーオプション」227 ページを参照してください。 |
| **-f** | トランザクションログなしでデータベースを強制的に起動します。「-f dbeng12/dbsrv12 サーバーリカバリオプション」228 ページを参照してください。 |
| **-fc** *filename* | ファイルシステムフルのコールバック関数を含む DLL のファイル名を指定します。「-fc dbeng12/dbsrv12 サーバーオプション」229 ページを参照してください。 |
| **-fips** | データベースおよび通信の暗号化に FIPS 認定のアルゴリズムの使用を要求します (Windows、UNIX、Linux)。「-fips dbeng12/dbsrv12 サーバーオプション」230 ページを参照してください。 |
| **-ga** | 最後の非 HTTP クライアント接続を閉じた後、データベースを自動的にアンロードします。さらに、最後のデータベースを閉じた後で停止します。「-ga dbeng12/dbsrv12 サーバーオプション」231 ページを参照してください。 |
| **-gb** *level* | データベースプロセスの優先度クラスを *level* に設定します (Windows、UNIX、Mac OS X)。「-gb dbeng12/dbsrv12 サーバーオプション」232 ページを参照してください。 |
| **-gc** *num* | 最大チェックポイントタイムアウト時間を *num* 分に設定します。「-gc dbeng12/dbsrv12 サーバーオプション」232 ページを参照してください。 |
| **-gd** *level* | データベース起動パーミッションを設定します。「-gd dbeng12/dbsrv12 サーバーオプション」233 ページを参照してください。 |
| **-ge** *size* | 外部関数を実行するスレッドのスタックサイズを設定します。「-ge dbeng12/dbsrv12 サーバーオプション」234 ページを参照してください。 |
| **-gf** | トリガーの起動を無効にします。「-gf dbeng12/dbsrv12 サーバーオプション」235 ページを参照してください。 |
| **-gk** *level* | サーバーを停止するためのパーミッションを設定します。「-gk dbeng12/dbsrv12 サーバーオプション」235 ページを参照してください。 |
| **-gl** *level* | データをロードまたはアンロードするためのパーミッションを設定します。「-gl dbeng12/dbsrv12 サーバーオプション」236 ページを参照してください。 |

| サーバーオプション | 説明 |
|---|---|
| **-gm** *num* | 接続の最大数を設定します。「-gm dbeng12/dbsrv12 サーバーオプション」237 ページを参照してください。 |
| **-gn** *num* | データベースサーバーのマルチプログラミングレベルを設定します。「-gn dbsrv12 サーバーオプション」237 ページを参照してください。 |
| **-gna** | データベースサーバーのマルチプログラミングレベルの自動チューニングを制御します。「-gna dbsrv12 サーバーオプション」238 ページを参照してください。 |
| **-gnh** *num* | データベースサーバーが同時に実行できるタスクの最大数を設定します。「-gnh dbsrv12 サーバーオプション」239 ページを参照してください。 |
| **-gnl** *num* | データベースサーバーが同時に実行できるタスクの最小数を設定します。「-gnl dbsrv12 サーバーオプション」240 ページを参照してください。 |
| **-gns** | データベースサーバーメッセージログにマルチプログラミングレベルの統計をレポートします。「-gns dbsrv12 サーバーオプション」241 ページを参照してください。 |
| **-gp** *size* | 最大ページサイズを *size* バイトに設定します。「-gp dbeng12/dbsrv12 サーバーオプション」242 ページを参照してください。 |
| **-gr** *minutes* | リカバリの最大時間を設定します。「-gr dbeng12/dbsrv12 サーバーオプション」242 ページを参照してください。 |
| **-gss** *size* | スレッドスタックサイズを *size* バイトに設定します。「-gss dbeng12/dbsrv12 サーバーオプション」243 ページを参照してください。 |
| **-gt** *num* | 使用できる物理プロセッサーの最大数を設定します (ライセンスされたプロセッサーの数を上限とする)。このオプションは、マルチプロセッサーシステムでのみ役立ちます。「-gt dbeng12/dbsrv12 サーバーオプション」244 ページを参照してください。 |
| **-gtc** *logical-processors-to-use* | データベースサーバーが許容するプロセッサー同時実行性の最大値を指定します。「-gtc dbeng12/dbsrv12 サーバーオプション」245 ページを参照してください。 |
| **-gu** *level* | ユーティリティコマンドのパーミッションレベルを utility_db、all、none、DBA のいずれかに設定します。「-gu dbeng12/dbsrv12 サーバーオプション」247 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| サーバーオプション | 説明 |
|---|---|
| **-im** *submode* | データベースサーバーをインメモリで実行し、ディスクへの書き込みをなくしたり軽減したりします。「-im dbeng12/dbsrv12 サーバーオプション」247 ページを参照してください。 |
| **-k** | パフォーマンスモニター統計値の収集を制御します。「-k dbeng12/dbsrv12 サーバーオプション」249 ページを参照してください。 |
| **-kl** *GSS-API-library-file* | Kerberos GSS-API ライブラリ (UNIX では共有オブジェクト) のファイル名を指定し、データベースサーバーへの Kerberos 認証接続を有効にします。「-kl dbeng12/dbsrv12 サーバーオプション」250 ページを参照してください。 |
| **-kp** *server-principal* | Kerberos サーバープリンシパルを指定し、データベースサーバーへの Kerberos 認証接続を有効にします。「-kp dbeng12/dbsrv12 サーバーオプション」251 ページを参照してください。 |
| **-kr** *server-realm* | Kerberos サーバープリンシパルの領域を指定し、データベースサーバーへの Kerberos 認証接続を有効にします。「-kr dbeng12/dbsrv12 サーバーオプション (旧式)」252 ページを参照してください。 |
| **-krb** | データベースサーバーへの Kerberos 認証接続を有効にします。「-krb dbeng12/dbsrv12 サーバーオプション」253 ページを参照してください。 |
| **-ks** | データベースサーバーからカウンター値を収集するためにパフォーマンスモニターで使用される共有メモリの作成を無効にします (Windows)。「-ks dbeng12/dbsrv12 サーバーオプション」254 ページを参照してください。 |
| **-ksc** | パフォーマンスモニターでモニターできる接続の最大数を指定します (Windows)。「-ksc dbeng12/dbsrv12 サーバーオプション」255 ページを参照してください。 |
| **-ksd** | パフォーマンスモニターでモニターできるデータベースの最大数を指定します (Windows)。「-ksd dbeng12/dbsrv12 サーバーオプション」255 ページを参照してください。 |
| **-m** | すべてのデータベースについて、各チェックポイントの実行後にトランザクションログをトランケートします。「-m dbeng12/dbsrv12 サーバーオプション」255 ページを参照してください。 |
| **-n** *name* | データベースサーバー名を *name* にします。**-n オプションは指定する位置によって意味が異なります。** 「-n dbeng12/dbsrv12 サーバーオプション」257 ページを参照してください。 |

| サーバーオプション | 説明 |
|---|---|
| **-o** *filename* | 指定したファイルにメッセージを出力します。「-o dbeng12/dbsrv12 サーバーオプション」258 ページを参照してください。 |
| **-oe** *filename* | 起動エラー、致命的なエラー、アサーションをログするファイルを指定します。「-oe dbeng12/dbsrv12 サーバーオプション」259 ページを参照してください。 |
| **-on** *size* | データベースサーバーメッセージログファイルの最大サイズを指定します。ログファイルがこのサイズに達すると、現在のファイルが拡張子 *.old* の付いた名前に変更され、新しいファイルが作成されます。「-on dbeng12/dbsrv12 サーバーオプション」260 ページを参照してください。 |
| **-os** *size* | メッセージ用のログファイルのサイズを制限します。「-os dbeng12/dbsrv12 サーバーオプション」261 ページを参照してください。 |
| **-ot** *filename* | データベースサーバーメッセージログファイルをトランケートし、そのファイルに出力メッセージを追加します。「-ot dbeng12/dbsrv12 サーバーオプション」262 ページを参照してください。 |
| **-p** *packet-size* | 最大通信パケットサイズを設定します (ネットワークサーバー)。「-p dbeng12/dbsrv12 サーバーオプション」262 ページを参照してください。 |
| **-pc** | 同一コンピューター接続以外のすべての通信パケットを圧縮します。「-pc dbsrv12 サーバーオプション」263 ページを参照してください。 |
| **-pt** *size-in-bytes* | 圧縮を適用する最小のネットワークパケットサイズを設定します。「-pt dbsrv12 サーバーオプション」264 ページを参照してください。 |
| **-qi** | データベースサーバーシステムトレイアイコンまたはデータベースサーバーメッセージウィンドウを非表示にします (Windows)。「-qi dbeng12/dbsrv12 サーバーオプション」264 ページを参照してください。 |
| **-qn** | 起動時にデータベースサーバーメッセージウィンドウを最小化しません (Windows と Linux)。「-qn dbeng12/dbsrv12 サーバーオプション」265 ページを参照してください。 |
| **-qp** | データベースサーバーメッセージウィンドウにパフォーマンスに関するメッセージを表示しないようにします。「-qp dbeng12/dbsrv12 サーバーオプション」266 ページを参照してください。 |
| **-qs** | 起動エラーウィンドウを表示しないようにします (Windows)。「-qs dbeng12/dbsrv12 サーバーオプション」266 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| サーバーオプション | 説明 |
|---|---|
| **-qw** | データベースサーバーメッセージウィンドウを表示しないようにします。「-qw dbeng12/dbsrv12 サーバーオプション」267 ページを参照してください。 |
| **-r** | 読み込み専用モードでデータベースを開きます。「-r dbeng12/dbsrv12 サーバーオプション」267 ページを参照してください。 |
| **-s** *facility-ID* | Syslog facility ID を設定します (UNIX、Mac OS X)。「-s dbeng12/dbsrv12 サーバーオプション」268 ページを参照してください。 |
| **-sb** { **0** \| **1** } | ブロードキャストに対するサーバーの動作を指定します。「-sb dbeng12/dbsrv12 サーバーオプション」270 ページを参照してください。 |
| **-sf** *feature-list* | このデータベースサーバーで実行されるデータベースの機能を保護します。「-sf dbeng12/dbsrv12 サーバーオプション」271 ページを参照してください。 |
| **-sk** *key* | データベースサーバーで無効になっている機能を有効にするためのキーを指定します。「-sk dbeng12/dbsrv12 サーバーオプション」275 ページを参照してください。 |
| **-su** *password* | ユーティリティデータベース (utility_db) の DBA ユーザーのパスワードを設定します。または、ユーティリティデータベースへの接続を無効にします。「-su dbeng12/dbsrv12 サーバーオプション」276 ページを参照してください。 |
| **-ti** *minutes* | シャットダウンするまでのクライアントのアイドル時間を設定します (デフォルト値は 240 分)。「-ti dbeng12/dbsrv12 サーバーオプション」277 ページを参照してください。 |
| **-tl** *seconds* | デフォルトのクライアント活性タイムアウト (秒数) を設定します (デフォルト値は 120 秒)。「-tl dbeng12/dbsrv12 サーバーオプション」278 ページを参照してください。 |
| **-tmf** | トランザクションマネージャーに、強制的に分散トランザクションをリカバリさせます (Windows)。「-tmf dbeng12/dbsrv12 サーバーオプション」279 ページを参照してください。 |
| **-tmt** *milliseconds* | 分散トランザクションの再エンリストタイムアウトを設定します (Windows)。「-tmt dbeng12/dbsrv12 サーバーオプション」280 ページを参照してください。 |
| **-tq** *time* | 終了時刻を設定します (ネットワークサーバー)。「-tq dbeng12/dbsrv12 サーバーオプション」280 ページを参照してください。 |

| サーバーオプション | 説明 |
|---|---|
| **-u** | バッファーディスク I/O を使用します (Windows、UNIX、Mac OS X)。「-u dbeng12/dbsrv12 サーバーオプション」281 ページを参照してください。 |
| **-ua** | 非同期 I/O の使用をオフにします (Linux)。「-ua dbeng12/dbsrv12 サーバーオプション」281 ページを参照してください。 |
| **-uc** | データベースサーバーをシェルモードで起動します (UNIX と Mac OS X)。「-uc dbeng12/dbsrv12 サーバーオプション」282 ページを参照してください。 |
| **-ud** | デーモンとして実行します (UNIX、Mac OS X)。「-ud dbeng12/dbsrv12 サーバーオプション」282 ページを参照してください。 |
| **-uf** | 致命的なエラーの発生時に実行するアクションを指定します (UNIX、Mac OS X)。「-uf dbeng12/dbsrv12 サーバーオプション」283 ページを参照してください。 |
| **-ui** | **[サーバー起動オプション]** ウィンドウを開いてデータベースサーバーメッセージウィンドウを表示するか、使用可能な表示がない場合はデータベースサーバーをシェルモードで起動します (Linux と Mac OS X)。「-ui dbeng12/dbsrv12 サーバーオプション」284 ページを参照してください。 |
| **-um** | **[サーバー起動オプション]** ウィンドウを開き、データベースサーバーメッセージウィンドウを表示します (Mac OS X)。「-um dbeng12/dbsrv12 サーバーオプション」284 ページを参照してください。 |
| **-ut** *minutes* | *min* 分ごとにテンポラリファイルにタッチします (UNIX、Mac OS X)。「-ut dbeng12/dbsrv12 サーバーオプション」285 ページを参照してください。 |
| **-ux** | データベースサーバーメッセージウィンドウと **[サーバー起動オプション]** ウィンドウを表示します (Linux)。「-ux dbeng12/dbsrv12 サーバーオプション」286 ページを参照してください。 |
| **-v** | データベースサーバーのバージョンを表示して停止します。「-v dbeng12/dbsrv12 サーバーオプション」287 ページを参照してください。 |
| **-vss**{ + \| - } | ボリュームシャドウコピーサービス (VSS) を有効または無効にします (Windows)。「-vss dbeng12/dbsrv12 サーバーオプション」287 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| サーバーオプション | 説明 |
|---|---|
| **-wc**[ + \| - ] | データベースサーバーで実行されるデータベースの書き込みチェックサムを有効にします。「-wc dbeng12/dbsrv12 サーバーオプション」288 ページを参照してください。 |
| **-x** *list* | カンマで区切られた使用する通信プロトコルのリストを指定します。「-x dbeng12/dbsrv12 サーバーオプション」289 ページを参照してください。 |
| **-xa** *authentication-info* | 監視サーバーに対するデータベース名と認証文字列のリストを指定します。「-xa dbsrv12 サーバーオプション」290 ページを参照してください。 |
| **-xd** | データベースサーバーがデフォルトのデータベースサーバーにならないようにします。「-xd dbeng12/dbsrv12 サーバーオプション」292 ページを参照してください。 |
| **-xf** *state-file* | データベースミラーリングシステムに関するステータス情報の管理に使用されるファイルのロケーションを指定します。「-xf dbsrv12 サーバーオプション」292 ページを参照してください。 |
| **-xm** *seconds* | 新しい IP アドレスをチェックする時間を秒単位で設定します。最小値は 10、デフォルト値は 0 です。ポータブルデバイスの場合、デフォルト値は 120 です。「-xm dbeng12/dbsrv12 サーバーオプション」293 ページを参照してください。 |
| **-xs** | サーバー側の Web サービス通信プロトコルを指定します。「-xs dbeng12/dbsrv12 サーバーオプション」294 ページを参照してください。 |
| **-z** | 通信リンクに関する診断情報を提供します (ネットワークサーバー)。「-z dbeng12/dbsrv12 サーバーオプション」296 ページを参照してください。 |
| **-ze** | データベースサーバー環境変数をデータベースサーバーメッセージウィンドウに表示します。「-ze dbeng12/dbsrv12 オプション」297 ページを参照してください。 |
| **-zl** | 各接続の最後に作成された SQL 文の取得をオンにします。「-zl dbeng12/dbsrv12 サーバーオプション」297 ページを参照してください。 |
| **-zn** *integer* | 保持する要求ログファイルのコピー数を指定します。「-zn dbeng12/dbsrv12 サーバーオプション」298 ページを参照してください。 |

| サーバーオプション | 説明 |
|---|---|
| **-zo** *filename* | 要求ロギング情報を別個のファイルにリダイレクトします。「-zo dbeng12/dbsrv12 サーバーオプション」299 ページを参照してください。 |
| **-zoc** | Web サービスクライアント情報をファイルにリダイレクトします。「-zoc dbeng12/dbsrv12 サーバーオプション」300 ページを参照してください。 |
| **-zp** | クエリオプティマイザーが最近使用したプランの取得をオンにします。「-zp dbeng12/dbsrv12 サーバーオプション」301 ページを参照してください。 |
| **-zr** { **all** \| **SQL** \| **none** } | SQL オペレーションのログを有効にします。デフォルトは、NONE です。「-zr dbeng12/dbsrv12 サーバーオプション」301 ページを参照してください。 |
| **-zs** *size* | 要求ロギング用のログファイルのサイズを制限します。「-zs dbeng12/dbsrv12 サーバーオプション」303 ページを参照してください。 |
| **-zt** | 要求タイミング情報のロギングをオンにします。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |

## データベースオプション

次のオプションは、データベースサーバーコマンドのデータベースファイル名の後にだけ指定できます。

| データベースオプション | 説明 |
|---|---|
| **-a** *filename* | 名前付きトランザクションログファイルを適用します。「-a dbeng12/dbsrv12 データベースオプション」305 ページを参照してください。 |
| **-ad** *log-directory* | データベースに適用されるトランザクションログファイルがあるディレクトリを指定します。「-ad dbeng12/dbsrv12 データベースオプション」306 ページを参照してください。 |
| **-ar** | トランザクションログと同じディレクトリ内にあるログファイルをデータベースに適用します。「-ar dbeng12/dbsrv12 データベースオプション」307 ページを参照してください。 |
| **-as** | トランザクションログの適用後もデータベースの実行を継続します (-ad または -ar とともに使用)。「-as dbeng12/dbsrv12 データベースオプション」308 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| データベースオプション | 説明 |
|---|---|
| **-dh** | このサーバーに対して dblocate が実行された場合、データベースを表示しません。「-dh dbeng12/dbsrv12 データベースオプション」309 ページを参照してください。 |
| **-ds** | データベースの DB 領域のロケーションを指定します。「-ds dbeng12/dbsrv12 データベースオプション」308 ページを参照してください。 |
| **-ek** *key* | 暗号化キーを指定します。「-ek dbeng12/dbsrv12 データベースオプション」310 ページを参照してください。 |
| **-m** | 指定のデータベースについて、各チェックポイントの実行後にトランザクションログをトランケート (削除) します。「-m dbeng12/dbsrv12 データベースオプション」311 ページを参照してください。 |
| **-n** *name* | データベースに名前を付けます。**-n オプションは指定する位置によって意味が異なります。** 「-n dbeng12/dbsrv12 データベースオプション」312 ページを参照してください。 |
| **-r** | 読み込み専用モードで指定のデータベースを開きます。データベースは変更できません。「-r dbeng12/dbsrv12 データベースオプション」313 ページを参照してください。 |
| **-sm** | 読み込み専用のミラーデータベースへのアクセスに使用できるデータベースサーバー名を指定します。「-sm dbsrv12 データベースオプション (旧式)」314 ページを参照してください。 |
| **-sn** *alternate-server-name* | データベースサーバー上で動作する 1 つのデータベースに代替サーバー名を割り当てます。「-sn dbsrv12 オプション」315 ページを参照してください。 |
| **-wc**[ + \| - ] | データベースサーバーで実行されるデータベースの書き込みチェックサムを有効にします。「-wc dbeng12/dbsrv12 データベースオプション」317 ページを参照してください。 |
| **-xp** *mirroring-options* | データベースミラーリングが使用されている場合に、稼働しているサーバーにパートナーと監視サーバーへの接続を可能にする情報を提供します。「-xp dbsrv12 データベースオプション」318 ページを参照してください。 |

## 備考

データベースサーバーのコマンドの要素には、次のようなものがあります。

- **実行プログラム** **dbeng12** コマンドは、パーソナルデータベースサーバーを起動します。**dbsrv12** コマンドは、ネットワークデータベースサーバーを起動します。

1 つの例外を除いて、サポートされている各オペレーティングシステムで、パーソナルデータベースサーバーとネットワークデータベースサーバーの両方が提供されています。Windows Mobile では、ネットワークデータベースサーバーのみ提供されます。ネットワークサーバーでは TCP/IP がサポートされるため、使用しているデスクトップコンピューターから Sybase Central によってデータベース管理などのタスクを実行できます。

Windows Mobile 以外の Windows オペレーティングシステムでは、パーソナルデータベースサーバーの実行プログラムの名前は *dbeng12.exe* です。UNIX オペレーティングシステムでは、dbeng12 です。

Windows Mobile を含む Windows オペレーティングシステムでは、パーソナルデータベースサーバーの実行プログラムの名前は *dbsrv12.exe* です。Linux と UNIX オペレーティングシステムでは、dbsrv12 です。

- **サーバーオプション**　これらのオプションは、実行中のすべてのデータベースに対するデータベースサーバーの動作を制御します。
- **データベースファイル**　1 つまたは複数のデータベースファイル名を指定するか、まったく指定しないこともできます。指定された各データベースが起動され、引き続きアプリケーションで使用できます。

> **警告**
> データベースファイルとトランザクションログファイルは、データベースサーバーと同じ物理コンピューターに保存してください。または SAN や iSCSI 設定でアクセスできるようにしてください。リモートネットワークディレクトリにデータベースファイルやトランザクションログファイルを配置すると、パフォーマンスが低下したり、データが破壊されたり、データベースサーバーが不安定になったりする可能性があります。
>
> 詳細については、http://www.ianywhere.jp/tech/techdoc.html#s93 を参照してください。
>
> 最適な結果を得るために、トランザクションログは、データベースファイルとは別のディスクに保存してください。「トランザクションログ」127 ページを参照してください。

- **データベースオプション**　開始するデータベースファイルごとに、その動作の特定の状態を制御するデータベースオプションを指定できます。

データベースオプションとサーバーオプションでは、通常は大文字と小文字が区別されます。オプションはすべて小文字で入力してください。

*database-file* には、データベースファイル名を指定します。*database-file* にファイル拡張子のない名前が指定されると、データベースサーバーは *database-file* に拡張子 *.db* を付けてファイルを検索します。相対パスを使用する場合、そのパスは現在の作業フォルダーと相対関係となります。フルパスも指定できます。

オプションもデータベースファイルも指定しなかった場合、Windows オペレーティングシステムではウィンドウが表示され、データベースファイルを検索できます。

データベースサーバーをバッチファイルから起動する場合は、dbspawn ユーティリティを使用します。「サーバーバックグラウンド起動ユーティリティ (dbspawn)」1022 ページを参照してください。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

パーソナルデータベースサーバーでは最大 10 の同時接続を使用でき、要求処理には最大 1 つの CPU を使用し、ネットワーククライアント／サーバー接続はサポートしません。デフォルトでは、パーソナルデータベースサーバーは共有メモリプロトコルのみを使用します。パーソナルデータベースサーバーで TCP/IP を使用する場合は、-x オプションを使用する必要があります。「-x dbeng12/dbsrv12 サーバーオプション」289 ページを参照してください。

さらに、新規データベースを開始するために必要なデフォルトのパーミッションレベルや、CHECKPOINT 文を実行するために必要なパーミッションなどの小さな相違点もあります。パーソナルデータベースサーバーとネットワークデータベースサーバーの違いについては、「SQL Anywhere データベースサーバーの使用」141 ページを参照してください。

デフォルトでは、データベースサーバーページサイズは、コマンドラインで指定されたデータベースの最大ページサイズと同じ大きさです。データベースサーバーがいったん起動すると、データベースサーバーページより大きいページサイズのデータベースを起動することはできません。「最大ページサイズの考慮事項」150 ページを参照してください。

**参照**

● 「データベースサーバーの起動例」145 ページ

**例**

次のコマンドは、パーソナルデータベースサーバーで SQL Anywhere サンプルデータベースを起動します。

```
dbeng12 "%SQLANYSAMP12%¥demo.db"
```

次のコマンドは、ネットワークデータベースサーバーで SQL Anywhere サンプルデータベースを起動します。

```
dbsrv12 "%SQLANYSAMP12%¥demo.db"
```

次の例 (すべて 1 行で入力) は、**myserver** という名前のデータベースサーバーをキャッシュサイズ 3 MB で起動し、サンプルデータベースをロードします。

```
dbeng12 -c 3m -n myserver "%SQLANYSAMP12%¥demo.db"
```

# データベースサーバーオプション

次のオプションは、個々のデータベースだけでなく、データベースサーバー全体に適用されます。

## @data dbeng12/dbsrv12 サーバーオプション

指定された環境変数または設定ファイルからオプションを読み込みます。

**構文**

{ **dbeng12** | **dbsrv12** } **@***data* ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー (Windows Mobile を除く)。言語選択ユーティリティ (dblang)、証明書の作成ユーティリティ (createcert)、証明書ビューアーユーティリティ (viewcert)、Microsoft ActiveSync プロバイダーインストールユーティリティ (mlasinst)、ファイル非表示ユーティリティ (dbfhide) を除くすべてのデータベースユーティリティでサポートされます。

- 言語選択ユーティリティ (dblang)
- 証明書作成ユーティリティ (createcert)
- 証明書ビューアーユーティリティ (viewcert)
- Microsoft ActiveSync プロバイダーインストールユーティリティ (mlasinst)
- ファイル難読化ユーティリティ (dbfhide)

### 備考

このオプションを使用して、指定された環境変数または設定ファイルからコマンドラインオプションを読み出します。指定された名前と同じ名前の環境変数と設定ファイルが両方存在する場合は、環境変数が使用されます。

設定ファイルには、改行を含めたり、あらゆるオプションの設定を格納したりできます。「設定ファイル」909 ページを参照してください。

設定ファイルの情報を (パスワードが含まれるなどの理由で) 保護する場合は、ファイル難読化ユーティリティ (dbfhide) を使用して、設定ファイルの内容を難読化できます。

@data パラメーターはコマンドの任意の位置に指定でき、ファイルに含まれるパラメーターがその位置に挿入されます。複数のファイルを指定可能で、ファイル指定子をコマンドラインオプションで使用できます。

### 参照


- 「設定ファイル」909 ページ
- 「ファイル非表示ユーティリティ (dbfhide)」953 ページ


### 例

次の設定ファイルには、**myserver** という名前のサーバーをキャッシュサイズ 4 MB で起動し、サンプルデータベースをロードするオプションのセットが含まれています。

```
-c 4096
-n myserver
"c:¥mydatabase.db"
```

この設定ファイルを *c:¥config.txt* として保存すると、コマンドで次のように使用できます。

```
dbsrv12 @c:¥config.txt
```

次の設定ファイルにはコメントが含まれています。

```
#This is the server name:
-n MyServer
#These are the protocols:
-x tcpip
#This is the database file
my.db
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

次の文は、データベースサーバーをキャッシュサイズ 4 MB で起動し、サンプルデータベースをロードするオプションを格納する環境変数を設定します。

```
SET envvar=-c 4096 "c:¥mydatabase.db";
```

次のコマンドは、**envvar** という環境変数を使用してデータベースサーバーを起動します。

```
dbsrv12 @envvar
```

## -? dbeng12/dbsrv12 サーバーオプション

使用法を表示します。

### 構文

**{ dbeng12 | dbsrv12 } -?**

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー (Windows Mobile を除く)。

### 備考

このオプションを指定すると、各サーバーオプションの簡単な説明が表示されます。

## -b dbeng12/dbsrv12 サーバーオプション

バルクオペレーションモードを使用します。

### 構文

**{ dbeng12 | dbsrv12 } -b** ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

このオプションは、Interactive SQL INPUT 文を使用して、大量のデータをデータベースにロードする場合に役立ちます。

LOAD TABLE を使用してデータをバルクロードする場合は、-b オプションは使用しないでください。

このオプションを使用した場合、データベースサーバーでは、1 つのアプリケーションで 1 つの接続しか確立できなくなります。ロールバックログは保存されますが、トランザクションログは保存されません。マルチユーザーロッキングメカニズムはオフになっています。

-b オプションでデータをロードした後、初めてデータベースサーバーを起動する場合は、新しいトランザクションログファイルを使用してください。

バルクオペレーションモードにしても、トリガーは起動不可になりません。

### 参照

- 「INPUT 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「LOAD TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「バルクオペレーションのデータリカバリの問題」『SQL Anywhere サーバー SQL の使用法』
- 「バルクオペレーションのパフォーマンスの側面」『SQL Anywhere サーバー SQL の使用法』

## -c dbeng12/dbsrv12 サーバーオプション

データベースページや他のデータベースサーバー情報をキャッシュするために予約される初期メモリを設定します。

### 構文

**{ dbeng12 | dbsrv12 } -c {** *size*[ **k | m | g | p** ] **}** ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

データベースサーバーキャッシュで使用できるキャッシュメモリの量は、パフォーマンスを制御する主要な要因の 1 つになります。初期のキャッシュメモリ量は、-c サーバーオプションを使用して設定できます。データベースサーバー用のキャッシュメモリが大きいほど、パフォーマンスは向上します。

*size* には、メモリ容量をバイト単位で指定します。単位をキロバイト、メガバイト、またはギガバイトで指定するには、それぞれ **k**、**m**、または **g** を使用してください。

単位 **p** は、合計物理システムメモリと非 AWE の最大キャッシュサイズのうち、いずれか小さい方のパーセンテージを表します。非 AWE の最大キャッシュサイズは、オペレーティングシステムによって異なります。次に例を示します。

- 2.5 GB － Windows 32 ビット Advanced Server、Enterprise Server、Datacenter Server、Vista
- 3.5 GB － Windows x64 Edition 上の 32 ビットデータベースサーバー
- 1.5 GB － その他すべての 32 ビット Windows オペレーティングシステム
- Windows Mobile の場合、キャッシュサイズは使用可能な物理メモリによって制限される
- 64 ビットのデータベースサーバーの場合、キャッシュサイズは無制限と見なされる

**p** を使用すると、引数はパーセンテージを表します。**p** の代わりに % も使用できますが、Windows オペレーティングシステムではほとんどの場合 % を環境変数のエスケープ文字として使用するため、% 文字をエスケープする必要があります。たとえば、初期キャッシュサイズを物理システムメモリの 50% に設定するには、次のコマンドを実行します。

```
dbeng12 -c 50%% ...
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

-c オプションを指定しないと、初期キャッシュメモリの割り当てサイズが次のように計算されます。

- **Windows Mobile**　式は次のとおりです。

  max( 600 KB, min( dbsize, 0.25*TotalPhysicalMemory ) );

  *dbsize* は起動したデータベースファイルの合計サイズです。*TotalPhysicalMemory* は、コンピューターの物理メモリの合計量です。

- **Windows**　式は次のとおりです。

  max( 2 MB, min( dbsize, 0.25*TotalPhysicalMemory ) );

  *dbsize* は起動したデータベースファイルの合計サイズです。*TotalPhysicalMemory* は、コンピューターの物理メモリの合計量です。

  Windows で AWE キャッシュを使用している場合、式は次のとおりです。

  min( 100% of AvailablePhysicalMemory-128MB, dbsize );

  この値が 2 MB より小さい場合、AWE キャッシュは使用されていません。

  AWE キャッシュについては、「-cw dbeng12/dbsrv12 サーバーオプション (旧式)」217 ページを参照してください。

  **注意**
  AWE は使用されなくなりました。サイズの大きいキャッシュが必要な場合は、64 ビット版の Windows オペレーティングシステムで 64 ビット版のデータベースサーバーを使用することをおすすめします。

- **UNIX**　最小で 8 MB です。

  max( 8 MB, min( 0.1*( physical-memory + available-swap ) , database-size*1.1 )

  UNIX の初期キャッシュサイズについては、「動的キャッシュサイズ決定 (UNIX)」『SQL Anywhere サーバー SQL の使用法』を参照してください。

**注意**
初期キャッシュサイズまたは最小キャッシュサイズを最大キャッシュサイズの 1/8 よりも小さな値に設定しようとすると、初期キャッシュサイズおよび最小キャッシュサイズは最大キャッシュサイズに比例して自動的に増加します。

動的なキャッシュサイズの変更を無効 (-ca オプション) にすると、使用されるキャッシュサイズが使用可能なメモリのサイズによって制限されることがあります。「パフォーマンス向上のためのキャッシュの使用」『SQL Anywhere サーバー SQL の使用法』を参照してください。

データベースサーバーメッセージウィンドウには起動時のキャッシュサイズが表示されます。また、次の文を使用して現在のキャッシュサイズを取得することもできます。

```
SELECT PROPERTY( 'CurrentCacheSize' );
```

### 参照


- キャッシュサイズ 156 ページ
- 「動的キャッシュサイズ決定」『SQL Anywhere サーバー SQL の使用法』
- 「-ca dbeng12/dbsrv12 サーバーオプション」206 ページ
- 「-cc dbeng12/dbsrv12 サーバーオプション」207 ページ
- 「-ch dbeng12/dbsrv12 サーバーオプション」208 ページ
- 「-chx dbeng12/dbsrv12 サーバーオプション」209 ページ
- 「-cl dbeng12/dbsrv12 サーバーオプション」211 ページ
- 「-cm dbeng12/dbsrv12 サーバーオプション」212 ページ
- 「-cr dbeng12/dbsrv12 サーバーオプション」214 ページ
- 「-cs dbeng12/dbsrv12 サーバーオプション」215 ページ
- 「-cv dbeng12/dbsrv12 サーバーオプション」216 ページ
- 「-cw dbeng12/dbsrv12 サーバーオプション (旧式)」217 ページ
- 「パフォーマンス向上のためのキャッシュの使用」『SQL Anywhere サーバー SQL の使用法』


### 例

次の例は、キャッシュサイズ 3 MB で始まる **myserver** という名前のデータベースサーバーを起動し、サンプルデータベースをロードします。

```
dbeng12 -c 3m -n myserver "%SQLANYSAMP12%¥demo.db"
```

## -ca dbeng12/dbsrv12 サーバーオプション

静的キャッシュサイズを強制的に適用します。

### 構文

**{ dbeng12 | dbsrv12 } -ca 0** ...

### 適用対象

Windows、UNIX、Mac OS X

### 備考

-ca 0 オプションを指定すると、自動キャッシュサイズチューニングに無効にすることができます。-ca 0 オプションの設定がない場合、データベースサーバーは自動的にキャッシュサイズを増加します。このオプションを指定した場合、動的メモリが足りなくなったことを示すエラーがデータベースサーバーで発生しても、キャッシュサイズは引き続き調整されます。

このサーバーオプションは、-ca 0 という形式でのみ使用してください。

AWE キャッシュを使用している場合、このオプションは無視されます。「-cw dbeng12/dbsrv12 サーバーオプション (旧式)」217 ページを参照してください。

> **注意**
> AWE は使用されなくなりました。サイズの大きいキャッシュが必要な場合は、64 ビット版の Windows オペレーティングシステムで 64 ビット版のデータベースサーバーを使用することをおすすめします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照


- キャッシュサイズ 156 ページ
- 「動的キャッシュサイズ決定」『SQL Anywhere サーバー SQL の使用法』
- 「-c dbeng12/dbsrv12 サーバーオプション」204 ページ
- 「-cc dbeng12/dbsrv12 サーバーオプション」207 ページ
- 「-ch dbeng12/dbsrv12 サーバーオプション」208 ページ
- 「-chx dbeng12/dbsrv12 サーバーオプション」209 ページ
- 「-cl dbeng12/dbsrv12 サーバーオプション」211 ページ
- 「-cm dbeng12/dbsrv12 サーバーオプション」212 ページ
- 「-cr dbeng12/dbsrv12 サーバーオプション」214 ページ
- 「-cs dbeng12/dbsrv12 サーバーオプション」215 ページ
- 「-cv dbeng12/dbsrv12 サーバーオプション」216 ページ
- 「-cw dbeng12/dbsrv12 サーバーオプション (旧式)」217 ページ


### 例

次の例は、総物理メモリ量の 40% の静的キャッシュがある myserver というデータベースサーバーを起動し、サンプルデータベースをロードします。データベースサーバーでは、キャッシュサイズは自動的に調整されません。

```
dbsrv12 -c 40P -ca 0 -n myserver "%SQLANYSAMP12%¥demo.db"
```

## -cc dbeng12/dbsrv12 サーバーオプション

次回にデータベースが起動されるときに、キャッシュウォーミングに使用するデータベースページに関する情報を収集します。

### 構文

**{ dbeng12 | dbsrv12 } -cc{ + | - }** ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

デフォルトでは、ページ収集はオンになっています。収集がオンになると、データベースサーバーは要求された各データベースページを追跡し続けます。ページの収集は、最大ページ数が収集されるか、データベースが停止されるか、または収集率が最小値を下回った場合に終了します。収集する最大ページ数を設定したり、収集率の値を指定したりすることはできません (この値は、キャッシュサイズとデータベースサイズに基づいています)。いったん収集が停止すると、要求されたページに関する情報はデータベースに記録されるので、それらのページは次回データベースが -cr オプションで開始されるときに、キャッシュの準備に使用できます。参照されたページの収集は、デフォルトではオンになっています。

### 参照


- 「-c dbeng12/dbsrv12 サーバーオプション」204 ページ
- 「-ca dbeng12/dbsrv12 サーバーオプション」206 ページ
- 「-ch dbeng12/dbsrv12 サーバーオプション」208 ページ
- 「-chx dbeng12/dbsrv12 サーバーオプション」209 ページ
- 「-cl dbeng12/dbsrv12 サーバーオプション」211 ページ
- 「-cm dbeng12/dbsrv12 サーバーオプション」212 ページ
- 「-cr dbeng12/dbsrv12 サーバーオプション」214 ページ
- 「-cs dbeng12/dbsrv12 サーバーオプション」215 ページ
- 「-cv dbeng12/dbsrv12 サーバーオプション」216 ページ
- 「-cw dbeng12/dbsrv12 サーバーオプション (旧式)」217 ページ
- 「キャッシュウォーミング」『SQL Anywhere サーバー SQL の使用法』


## -ch dbeng12/dbsrv12 サーバーオプション

自動キャッシュ増加機能を利用した際の最大キャッシュサイズ上限値を設定します。

### 構文

{ **dbeng12** | **dbsrv12** } **-ch** { *size*[ **k** | **m** | **g** | **p** ] } ...

### 適用対象

Windows、UNIX、OS X

### 備考

このオプションは、データベースサーバーが自動キャッシュ増加機能を実行するときに使用できる、キャッシュサイズの上限を設定します。デフォルトでは、上限値の概算は、非 AWE の最大キャッシュサイズとコンピューターの物理メモリの 90% のうち、いずれか低い方になります。

*size* には、メモリ容量をバイト単位で指定します。単位をキロバイト、メガバイト、またはギガバイトで指定するには、それぞれ **k**、**m**、または **g** を使用してください。

単位 **p** は、物理システムメモリと非 AWE の最大キャッシュサイズのうち、いずれか小さい方のパーセンテージを表します。非 AWE の最大キャッシュサイズは、オペレーティングシステムによって異なります。次に例を示します。

●2.5 GB － Windows 32 ビット Advanced Server、Enterprise Server、Datacenter Server

●3.5 GB － Windows x64 Edition 上の 32 ビットデータベースサーバー

●1.5 GB － その他すべての 32 ビット Windows オペレーティングシステム

●Windows Mobile の場合、キャッシュサイズは使用可能な物理メモリによって制限される

●64 ビットのデータベースサーバーの場合、キャッシュサイズは無制限と見なされる

**p** を使用すると、引数はパーセンテージを表します。p を使用すると、引数はパーセンテージを表します。p の代わりに % も使用できますが、Windows オペレーティングシステムでは % 文字

をエスケープしないと使用できません。初期キャッシュサイズを物理システムメモリの 50% に設定するには、次のコマンドを実行します。

64 ビットのオペレーティングシステムでは、-ch と -chx は同じですが、-ch を使用することをおすすめします。

```
dbeng12 -ch 50%% ...
```

AWE キャッシュを使用している場合、このオプションは無視されます。「-cw dbeng12/dbsrv12 サーバーオプション (旧式)」217 ページを参照してください。

**注意**
AWE は使用されなくなりました。サイズの大きいキャッシュが必要な場合は、64 ビット版の Windows オペレーティングシステムで 64 ビット版のデータベースサーバーを使用することをおすすめします。

### 参照


- キャッシュサイズ 156 ページ
- 「動的キャッシュサイズ決定」『SQL Anywhere サーバー SQL の使用法』
- 「-c dbeng12/dbsrv12 サーバーオプション」204 ページ
- 「-ca dbeng12/dbsrv12 サーバーオプション」206 ページ
- 「-cc dbeng12/dbsrv12 サーバーオプション」207 ページ
- 「-chx dbeng12/dbsrv12 サーバーオプション」209 ページ
- 「-cl dbeng12/dbsrv12 サーバーオプション」211 ページ
- 「-cm dbeng12/dbsrv12 サーバーオプション」212 ページ
- 「-cr dbeng12/dbsrv12 サーバーオプション」214 ページ
- 「-cs dbeng12/dbsrv12 サーバーオプション」215 ページ
- 「-cv dbeng12/dbsrv12 サーバーオプション」216 ページ


### 例

次の例は、silver という名前のデータベースサーバーを最大キャッシュサイズ 2 MB で起動し、サンプルデータベースをロードします。

```
dbeng12 -ch 2m -n silver "%SQLANYSAMP12%¥demo.db"
```

## -chx dbeng12/dbsrv12 サーバーオプション

32 ビットのデータベースサーバーで、自動キャッシュ増加機能を利用した際の最大キャッシュサイズ上限値を設定します。

### 構文

**{ dbeng12 | dbsrv12 } -chx** { *size*[ **k** | **m** | **g** | **p** ] } ...

### デフォルト

最低でも 512 MB のアドレス領域が、キャッシュ以外の使用に確保されます。

### 適用対象

32 ビット Windows と 32 ビット UNIX

### 備考

**警告**
追加のアドレス領域を使用することによって、サーバーがロードする必要のある DLL などのサーバーアドレス領域、リモートテーブルアクセス用の ODBC ドライバー、ネットワークパケットバッファー、外部ストアドプロシージャー、キャッシュ以外のメモリ割り付けを使用する他のコンポーネントに影響を及ぼさないことを確認するテストを実行した場合にのみ、-chx オプションを使用してください。

- サーバーがロードする必要のある DLL
- リモートテーブルアクセス用の ODBC ドライバー
- ネットワークパケットバッファー
- 外部ストアドプロシージャー
- キャッシュ以外のメモリ割り付け

サイズの大きいキャッシュが必要な場合は、64 ビット版のオペレーティングシステムで 64 ビット版のデータベースサーバーを使用することをおすすめします。

最低でも 512 MB のアドレス領域をキャッシュ以外の使用に確保するため、通常、データベースサーバーではキャッシュサイズが制限されます。確保される最大キャッシュサイズからキャッシュ以外に使用するアドレス領域を差し引いたサイズを指定する場合は、-chx オプションを使用します。このオプションが適用されるのは、32 ビットのデータベースサーバーのみです。サイズの大きいキャッシュを使用すると、データベースサーバーが不安定になる可能性があります。このオプションの使用には十分注意してください。

非 AWE の最大キャッシュサイズは、オペレーティングシステムによって異なります。次に例を示します。

- 2.5 GB － Windows 32 ビット Advanced Server、Enterprise Server、Datacenter Server、Vista

- 3.5 GB － Windows x64 Edition 上の 32 ビットデータベースサーバー

- 1.5 GB － その他すべての 32 ビット Windows オペレーティングシステム

- Windows Mobile の場合、キャッシュサイズは使用可能な物理メモリによって制限される

- 64 ビットのデータベースサーバーの場合、キャッシュサイズは無制限と見なされる

64 ビットのオペレーティングシステムでは、-ch と -chx は同じですが、-ch を使用することをおすすめします。

*size* には、メモリ容量をバイト単位で指定します。単位をキロバイト、メガバイト、またはギガバイトで指定するには、それぞれ **k**、**m**、または **g** を使用してください。

単位 **p** は、物理システムメモリと非 AWE の最大キャッシュサイズのうち、いずれか小さい方のパーセンテージを表します。**p** の代わりに % も使用できますが、Windows ではほとんどの場合 % を環境変数のエスケープ文字として使用するため、% 文字をエスケープする必要があります。

### 参照


- キャッシュサイズ 156 ページ
- 「-c dbeng12/dbsrv12 サーバーオプション」204 ページ
- 「-ca dbeng12/dbsrv12 サーバーオプション」206 ページ
- 「-cc dbeng12/dbsrv12 サーバーオプション」207 ページ
- 「-ch dbeng12/dbsrv12 サーバーオプション」208 ページ
- 「-cl dbeng12/dbsrv12 サーバーオプション」211 ページ
- 「-cm dbeng12/dbsrv12 サーバーオプション」212 ページ
- 「-cr dbeng12/dbsrv12 サーバーオプション」214 ページ
- 「-cs dbeng12/dbsrv12 サーバーオプション」215 ページ
- 「-cv dbeng12/dbsrv12 サーバーオプション」216 ページ
- 「-cw dbeng12/dbsrv12 サーバーオプション (旧式)」217 ページ
- 「パフォーマンス向上のためのキャッシュの使用」『SQL Anywhere サーバー SQL の使用法』


## -cl dbeng12/dbsrv12 サーバーオプション

動的キャッシュサイズ変更機能に対して最小キャッシュサイズを設定します。

### 構文

{ **dbeng12** | **dbsrv12** } **-cl** { *size*[ **k** | **m** | **g** | **p** ] } ...

### デフォルト

Windows では 2 MB

UNIX では 8 MB

### 適用対象

Windows、UNIX、Mac OS X

### 備考

このオプションは、キャッシュの下限値を設定します。-c を指定し、-cl を指定しないと、最小キャッシュサイズは初期キャッシュサイズ (-c 設定) に設定されます。-c と-cl をどちらも指定しないと、必要に応じてキャッシュを縮小できるように、最小キャッシュサイズは低い定数値に設定されます。Windows プラットフォームでは、この値は 2 MB になります。

*size* には、メモリ容量をバイト単位で指定します。単位をキロバイト、メガバイト、またはギガバイトで指定するには、それぞれ **k**、**m**、または **g** を使用してください。

単位 **p** は、物理システムメモリと非 AWE の最大キャッシュサイズのうち、いずれか小さい方のパーセンテージを表します。非 AWE の最大キャッシュサイズは、オペレーティングシステムによって異なります。次に例を示します。

●2.5 GB － Windows 32 ビット Advanced Server、Enterprise Server、Datacenter Server

●3.5 GB － Windows x64 Edition 上の 32 ビットデータベースサーバー

●1.5 GB － その他すべての 32 ビットシステム

●Windows Mobile の場合、キャッシュサイズは使用可能な物理メモリによって制限される

●64 ビットのデータベースサーバーの場合、キャッシュサイズは無制限と見なされる

**p** を使用すると、引数はパーセンテージを表します。**p** の代わりに % も使用できますが、Windows ではほとんどの場合 % を環境変数のエスケープ文字として使用するため、% 文字をエスケープする必要があります。

たとえば、初期キャッシュサイズを物理システムメモリの 50% に設定するには、次のコマンドを実行できます。

```
dbeng12 -cl 50%% ...
```

AWE キャッシュを使用している場合、このオプションは無視されます。「-cw dbeng12/dbsrv12 サーバーオプション (旧式)」217 ページを参照してください。

> **注意**
> 初期キャッシュサイズまたは最小キャッシュサイズを最大キャッシュサイズの 1/8 よりも小さな値に設定しようとすると、初期キャッシュサイズおよび最小キャッシュサイズは最大キャッシュサイズに比例して自動的に増加します。

**参照**


- 「動的キャッシュサイズ決定」『SQL Anywhere サーバー SQL の使用法』
- 「-c dbeng12/dbsrv12 サーバーオプション」204 ページ
- 「-ca dbeng12/dbsrv12 サーバーオプション」206 ページ
- 「-cc dbeng12/dbsrv12 サーバーオプション」207 ページ
- 「-ch dbeng12/dbsrv12 サーバーオプション」208 ページ
- 「-chx dbeng12/dbsrv12 サーバーオプション」209 ページ
- 「-cm dbeng12/dbsrv12 サーバーオプション」212 ページ
- 「-cr dbeng12/dbsrv12 サーバーオプション」214 ページ
- 「-cs dbeng12/dbsrv12 サーバーオプション」215 ページ
- 「-cv dbeng12/dbsrv12 サーバーオプション」216 ページ
- 「-cw dbeng12/dbsrv12 サーバーオプション (旧式)」217 ページ


**例**

次の例は、silver という名前のデータベースサーバーを最小キャッシュサイズ 5 MB で起動し、データベースファイル *example.db* をロードします。

```
dbeng12 -cl 5m -n silver "c:¥example.db"
```

## -cm dbeng12/dbsrv12 サーバーオプション

Windows において、Address Windowing Extensions (AWE) キャッシュに割り付けるアドレス領域のサイズを指定します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

> **注意**
> AWE は使用されなくなりました。サイズの大きいキャッシュが必要な場合は、64 ビット版の Windows オペレーティングシステムで 64 ビット版のデータベースサーバーを使用することをおすすめします。

### 構文

{ **dbeng12** | **dbsrv12** } **-cm** { *size*[ **k** | **m** | **g** | **p** ] } ...

### 適用対象

Windows

### 備考

サポートされるプラットフォーム上で AWE キャッシュを使用する場合、データベースサーバーは 512 MB を除くアドレス領域全体を使用してキャッシュメモリにアクセスします。アドレス領域のうち 512 MB は、サーバーがロードする必要のある DLL やキャッシュ以外のメモリ割り付けなどの目的に使用されます。ほとんどのシステムでは、このデフォルト設定で十分です。確保されるアドレス領域のサイズを拡大または縮小する必要がある場合は、-cm オプションを指定します。データベースサーバーが使用しているアドレス領域のサイズは、起動時にデータベースサーバーメッセージウィンドウに表示されます。

*size* には、メモリ容量をバイト単位で指定します。単位をキロバイト、メガバイト、またはギガバイトで指定するには、それぞれ **k**、**m**、または **g** を使用してください。

単位 **p** は、非 AWE の最大キャッシュサイズのパーセンテージを表します。**p** を使用すると、引数はパーセンテージを表します。**p** の代わりに % も使用できますが、Windows オペレーティングシステムではほとんどの場合 % を環境変数のエスケープ文字として使用するため、% 文字をエスケープする必要があります。AWE アドレス領域のキャッシュサイズを、データベースサーバープロセスで使用できるアドレス領域の 50% に設定するには、次のように指定します。

```
dbeng12 -cm 50%% ...
```

### 参照


- 「-c dbeng12/dbsrv12 サーバーオプション」204 ページ
- 「-ca dbeng12/dbsrv12 サーバーオプション」206 ページ
- 「-cc dbeng12/dbsrv12 サーバーオプション」207 ページ
- 「-ch dbeng12/dbsrv12 サーバーオプション」208 ページ
- 「-chx dbeng12/dbsrv12 サーバーオプション」209 ページ
- 「-cl dbeng12/dbsrv12 サーバーオプション」211 ページ
- 「-cr dbeng12/dbsrv12 サーバーオプション」214 ページ
- 「-cs dbeng12/dbsrv12 サーバーオプション」215 ページ
- 「-cv dbeng12/dbsrv12 サーバーオプション」216 ページ
- 「-cw dbeng12/dbsrv12 サーバーオプション (旧式)」217 ページ


## -cp dbeng12/dbsrv12 サーバーオプション

クラスを検索する追加の一連のディレクトリまたは JAR ファイルを指定します。

### 構文

{ **dbeng12** | **dbsrv12** } **-cp** *location*[ **;***location* ... ] ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

「クラウドに関する注意点」このオプションは、クラウドのサーバーでは完全にテストされていません。

### 備考

データベース内の Java で使用しているすべてのクラスと JAR ファイルはデータベース内にインストールすることをおすすめします。クラスと JAR ファイルをデータベース内に保存すると、データベースを簡単に別のコンピューターやオペレーティングシステムに移動できます。また、クラスと JAR ファイルをデータベース内にインストールすると、データベースサーバーのクラスローダーでクラスとリソースをデータベースからフェッチできるという利点もあります。その結果、データベース内の Java を使用している各接続で、これらのクラスの独自のインスタンスと、クラス内の静的変数の独自のコピーを使用できます。

ただし、クラスまたは JAR ファイルをシステムクラスローダーによってロードする必要がある場合は、java_class_path データベースオプションまたは -cp dbeng12/dbsrv12 サーバーオプションで指定できます。どちらのオプションを指定しても、Java VM の起動用にデータベースサーバーによって構築されるクラスパスにクラスと JAR ファイルが追加されます。java_class_path データベースオプションは、サーバーが複数のデータベースを実行していて、それぞれのデータベースにシステムクラスローダーによってロードする必要のある異なるセットの JAR およびディレクトリがあるときに便利です。-cp データベースサーバーオプションは、サーバー上のすべてのデータベースが同一のクラスまたは JAR ファイルを必要とするときに便利です。

### 参照


- 「データベースにおける Java」『SQL Anywhere サーバー プログラミング』
- 「データベースに Java クラスを格納する方法は？」『SQL Anywhere サーバー プログラミング』
- 「java_class_path option」632 ページ


## -cr dbeng12/dbsrv12 サーバーオプション

データベースが最後に実行されたときに収集した情報を使用して、キャッシュとデータベースページを再ロード (準備) します。

### 構文

{ **dbeng12** | **dbsrv12** } **-cr**{ **+** | **-** } ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 備考

データベースが最後に起動したときに参照されたページを使用して、データベースサーバーにキャッシュを準備するよう指示できます (ページ収集は -cc オプションを使用してオンにします)。キャッシュウォーミングは、デフォルトではオンになっています。データベースが起動されると、サーバーはデータベースをチェックして、データベースが最後に起動されたときに要求されたページの収集があるかどうかを確認します。データベースにこの情報が含まれている場合、前に参照したページがキャッシュにロードされます。

データベースが最後に起動されたときに参照されたページのキャッシュを準備することで、同じクエリまたは似たようなクエリがデータベースの起動のたびに実行されるような場合に、パフォーマンスを改善できます。

### 参照


- 「-cc dbeng12/dbsrv12 サーバーオプション」207 ページ
- 「-cl dbeng12/dbsrv12 サーバーオプション」211 ページ
- 「-cm dbeng12/dbsrv12 サーバーオプション」212 ページ
- 「-cs dbeng12/dbsrv12 サーバーオプション」215 ページ
- 「-cv dbeng12/dbsrv12 サーバーオプション」216 ページ
- 「-cw dbeng12/dbsrv12 サーバーオプション (旧式)」217 ページ
- 「キャッシュウォーミング」『SQL Anywhere サーバー SQL の使用法』


## -cs dbeng12/dbsrv12 サーバーオプション

動的キャッシュサイズ決定に関する統計をデータベースサーバーメッセージウィンドウに表示します。

### 構文

{ **dbeng12** | **dbsrv12** } **-cs** ...

### 適用対象

Windows、UNIX

### 備考

トラブルシューティング目的で、キャッシュサイズの調整方法を決めるためにデータベースサーバーで使用される統計が、データベースサーバーメッセージウィンドウに表示されます。

### 参照


- CacheSizingStatistics サーバープロパティ756 ページ
- 「-c dbeng12/dbsrv12 サーバーオプション」204 ページ
- 「-ca dbeng12/dbsrv12 サーバーオプション」206 ページ
- 「-cc dbeng12/dbsrv12 サーバーオプション」207 ページ
- 「-ch dbeng12/dbsrv12 サーバーオプション」208 ページ
- 「-chx dbeng12/dbsrv12 サーバーオプション」209 ページ
- 「-cl dbeng12/dbsrv12 サーバーオプション」211 ページ
- 「-cm dbeng12/dbsrv12 サーバーオプション」212 ページ
- 「-cr dbeng12/dbsrv12 サーバーオプション」214 ページ
- 「-cv dbeng12/dbsrv12 サーバーオプション」216 ページ
- 「-cw dbeng12/dbsrv12 サーバーオプション (旧式)」217 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「キャッシュウォーミング」『SQL Anywhere サーバー SQL の使用法』


## -cv dbeng12/dbsrv12 サーバーオプション

データベースサーバーメッセージウィンドウでのキャッシュウォーミングに関するメッセージの表示を制御します。

### 構文

**{ dbeng12 | dbsrv12 } -cv{ + | - } ...**

### デフォルト

キャッシュウォーミングメッセージは表示されません。

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

-cv+ が指定されると、次のいずれかのキャッシュウォーミングアクティビティが発生した場合に、データベースサーバーメッセージウィンドウにメッセージが表示されます。

- 要求されたページの収集が開始または停止する (-cc サーバーオプションで制御)
- ページの再ロードが開始または停止する (-cr サーバーオプションで制御)

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照


- 「-c dbeng12/dbsrv12 サーバーオプション」204 ページ
- 「-ca dbeng12/dbsrv12 サーバーオプション」206 ページ
- 「-cc dbeng12/dbsrv12 サーバーオプション」207 ページ
- 「-ch dbeng12/dbsrv12 サーバーオプション」208 ページ
- 「-cl dbeng12/dbsrv12 サーバーオプション」211 ページ
- 「-cm dbeng12/dbsrv12 サーバーオプション」212 ページ
- 「-cr dbeng12/dbsrv12 サーバーオプション」214 ページ
- 「-cs dbeng12/dbsrv12 サーバーオプション」215 ページ
- 「-cw dbeng12/dbsrv12 サーバーオプション (旧式)」217 ページ
- 「キャッシュウォーミング」『SQL Anywhere サーバー SQL の使用法』


### 例

次のコマンドは、データベースページの収集とページのロードをオンにして *mydatabase.db* というデータベースを起動し、これらのアクティビティに関するメッセージをデータベースサーバーメッセージウィンドウに表示します。

```
dbsrv12 -cc+ -cr+ -cv+ mydatabase.db
```

## -cw dbeng12/dbsrv12 サーバーオプション (旧式)

データベースサーバーキャッシュサイズの設定を目的とした Windows での Address Windowing Extensions (AWE) の使用を有効にします。

> **注意**
> AWE は使用されなくなりました。サイズの大きいキャッシュが必要な場合は、64 ビット版の Windows オペレーティングシステムで 64 ビット版のデータベースサーバーを使用することをおすすめします。

### 構文

{ **dbeng12** | **dbsrv12** } **-cw** ...

### 適用対象

Windows

### 備考

データベースサーバーキャッシュで使用できるキャッシュメモリの量は、パフォーマンスを制御する主要な要因の 1 つになります。Windows は Address Windowing Extensions をサポートしているため、-cw オプションを使用して、システムに搭載されている最大物理メモリを基にサイズの大きいキャッシュを利用できます。

AWE キャッシュは、64 ビットのデータベースサーバーではサポートされていません。

| オペレーティングシステム (32 ビット) | 最大キャッシュサイズ (非 AWE) | Windows がサポートする最大メモリサイズ |
|---|---|---|
| Windows XP Home Edition | 1.5 GB | 2 GB |
| Windows XP Professional | 1.5 GB | 4 GB |
| Windows Server 2003 Web | 1.5 GB | 2 GB |
| Windows Server 2003 Standard | 1.5 GB | 4 GB |
| Windows Server 2003 Enterprise | 2.5 GB[1] | 32 GB |
| Windows Server 2003 Datacenter | 2.5 GB[1] | 64 GB |
| Windows Vista Ultimate | 2.5 GB[1] | 4 GB |
| Windows Vista Enterprise | 2.5 GB[1] | 4 GB |
| Windows Vista Business | 2.5 GB[1] | 4 GB |
| Windows Vista Home Premium | 2.5 GB[1] | 4 GB |
| Windows Vista Home Basic | 2.5 GB[1] | 4 GB |
| Windows Vista Starter | 2.5 GB[1] | 1 GB |
| Windows 7 Ultimate | 2.5 GB[1] | 4 GB |
| Windows 7 Enterprise | 2.5 GB[1] | 4 GB |
| Windows 7 Professional | 2.5 GB[1] | 4 GB |
| Windows 7 Home Premium | 2.5 GB[1] | 4 GB |
| Windows 7 Home Basic | 2.5 GB[1] | 4 GB |
| Windows 7 Starter | 2.5 GB[1] | 1 GB |
| Windows Server 2008 Datacenter | 2.5 GB[1] | 64 GB |
| Windows Server 2008 Enterprise | 2.5 GB[1] | 64 GB |
| Windows Server 2008 Standard | 2.5 GB[1] | 4 GB |
| Windows Server 2008 Small Business Server | 2.5 GB[1] | 4 GB |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オペレーティングシステム (32 ビット) | 最大キャッシュサイズ (非 AWE) | Windows がサポートする最大メモリサイズ |
|---|---|---|
| Windows Web Server 2008 | 2.5 GB[1] | 4 GB |

[1] メモリが 2 GB ～ 16 GB の場合は、4 ギガバイトのチューニング (4GT) を有効にします。

- **Windows XP/2003**　**/3GB** オプションを、*boot.ini* ファイルの "[operating systems]" セクションにある Windows ブート行に追加します。

- **Windows Vista 以降**　管理者として次のコマンドを実行します。

```
bcdedit /set increaseuserva 3072
```

AWE キャッシュを使用している場合、システムで使用可能なほとんどの物理メモリをキャッシュに割り当てることができます。

非 AWE キャッシュを使用して必要なサイズのキャッシュを設定できる場合は、そちらをおすすめします。AWE キャッシュで割り当てられるメモリはデータベースサーバーでしか使用できないためです。つまり、データベースサーバーの実行中は、オペレーティングシステムやその他のアプリケーションは、データベースサーバーキャッシュに割り当てられたメモリを使用できません。AWE キャッシュでは、動的なキャッシュサイズの変更はサポートされていません。このため、AWE キャッシュを使用するときに、-ch または -cl オプションを指定してキャッシュサイズの上限と下限を設定しても無視されます。

デフォルトでは、アドレス領域のうち 512 MB がデータベースサーバーの AWE キャッシュ以外の目的に確保されます (アドレス領域とは、ある時点にプログラムがアクセスできるメモリの量をいいます)。通常はこの量で十分ですが、確保されたアドレス領域の量を -cm オプションを使用して変更してもかまいません。

Windows Vista と Windows 7 では、昇格されたデータベースサーバーのみ AWE メモリを使用できます。Windows Vista または Windows 7 でデータベースサーバーを自動的に起動する場合は、自動的に起動されるデータベースサーバーの実行プログラムが昇格するよう、接続文字列で ELEVATE=YES を指定する必要があります。「Elevate 接続パラメーター」340 ページを参照してください。

AWE キャッシュを使用してデータベースサーバーを起動するには、次の条件があります。

- Windows Vista と Windows 7 の場合は、管理者としてデータベースサーバーを実行する必要があります。

- システムで使用可能なメモリが少なくとも 130 MB 必要です。

- メモリが 2 GB ～ 16 GB の場合は、4 ギガバイトのチューニング (4GT) を有効にします。

  - **Windows XP/2003**　**/3GB** オプションを、*boot.ini* ファイルの "[operating systems]" セクションにある Windows ブート行に追加します。

  - **Windows Vista 以降**　管理者として次のコマンドを実行します。

```
bcdedit /set increaseuserva 3072
```

- メモリが 16 GB より多い場合は、4GT を有効にしません。
- メモリが最低でも 4 GB ある場合は、PAE を有効にします。
  - **Windows XP/2003**　**/PAE** オプションを、*boot.ini* ファイルの [operating systems] セクションにある Windows ブート行に追加します。
  - **Windows Vista 以降**　管理者として次のコマンドを実行します。

    ```
    bcdedit /set pae ForceEnable
    ```

- サーバーを実行中のユーザー ID に「メモリにページをロック」権限を付与します。ここでは、Windows XP での手順を説明します。
  1. 管理者として Windows にログオンします。
  2. **[コントロール パネル]** を開きます。
  3. **[管理ツール]** をダブルクリックします。
  4. **[ローカル セキュリティ ポリシー]** をダブルクリックします。
  5. 左ウィンドウ枠の **[ローカル ポリシー]** をクリックします。
  6. **[ユーザー権利の割り当て]** をダブルクリックします。
  7. 右ウィンドウ枠の **[メモリ内のページのロック]** ポリシーをダブルクリックします。
  8. **[ユーザーまたはグループの追加]** をクリックします。
  9. ユーザー名を入力し、**[OK]** をクリックします。
  10. **[メモリ内のページのロック]** ウィンドウで **[OK]** をクリックします。
  11. 開いているウィンドウをすべて閉じ、コンピューターを再起動して設定を有効にします。

-cw オプションと -c オプションを指定すると、データベースサーバーは次のように初期キャッシュを割り当てようとします。

1. AWE キャッシュは、-c オプションで指定されたキャッシュサイズよりも大きくなることはありません。-c オプションで指定された値が 2 MB より小さい場合、AWE は使用されません。
2. AWE キャッシュは、使用可能なすべての物理メモリから 128 MB を引いた容量を上回ることはありません。
3. AWE キャッシュは 2 MB より小さくなることはありません。この最小容量の物理メモリが使用可能でない場合、AWE キャッシュは使用されません。

-cw オプションを指定し、-c オプションを指定しないと、データベースサーバーは次のように初期キャッシュを割り当てようとします。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

1. AWE キャッシュは、使用可能なメモリのうち、128 MB をオペレーティングシステムで使用して、それ以外の容量を 100% 使用します。

2. AWE キャッシュは、コマンドラインで指定されたメインデータベースファイルの合計サイズを上回ることはありません。メインデータベースファイル以外の追加の DB 領域は、この計算の対象とはなりません。ファイルを指定しないと、この値は 0 になります。

3. AWE キャッシュは 2 MB より小さくなることはありません。この最小容量の物理メモリが使用可能でない場合、AWE キャッシュは使用されません。

サーバーが AWE キャッシュを使用する場合、キャッシュページサイズは 4 KB 以上となり、動的キャッシュサイズ決定は無効になります。

「パフォーマンス向上のためのキャッシュの使用」『SQL Anywhere サーバー SQL の使用法』を参照してください。

### 参照


- 「-c dbeng12/dbsrv12 サーバーオプション」204 ページ
- 「-ca dbeng12/dbsrv12 サーバーオプション」206 ページ
- 「-cc dbeng12/dbsrv12 サーバーオプション」207 ページ
- 「-ch dbeng12/dbsrv12 サーバーオプション」208 ページ
- 「-cl dbeng12/dbsrv12 サーバーオプション」211 ページ
- 「-cm dbeng12/dbsrv12 サーバーオプション」212 ページ
- 「-cr dbeng12/dbsrv12 サーバーオプション」214 ページ
- 「-cs dbeng12/dbsrv12 サーバーオプション」215 ページ
- 「-cv dbeng12/dbsrv12 サーバーオプション」216 ページ


### 例

次の例は、**myserver** という名前のデータベースサーバーをキャッシュサイズ 12 GB で起動し、データベース *c:¥test¥mydemo.db* をロードします。

```
dbeng12 -n myserver -c 12G -cw c:¥test¥mydemo.db
```

## -dt dbeng12/dbsrv12 サーバーオプション

テンポラリファイルを保存するディレクトリを指定します。

### 構文

{ **dbeng12** | **dbsrv12** } **-dt** *temp-file-dir* …

### 適用対象

すべてのサーバーおよびオペレーティングシステム (UNIX 上の共有メモリ接続を除く)

### 備考

データベースサーバーでは、次の 2 種類のテンポラリファイルが作成されます。

- **データベースサーバー関連のテンポラリファイル (すべてのプラットフォームで作成)**　-dt オプションでは、データベースサーバー関連のテンポラリファイルを保存するディレクトリを指定できます。このオプションを指定しないでデータベースサーバーを起動すると、データベースサーバーは以下の環境変数をこの通りの順序で調べ、テンポラリファイルの保存先とするディレクトリを決定します。

| Windows/Unix |
| --- |
| 1. SATMP |
| 2. TMP |
| 3. TMPDIR |
| 4. TEMP |

これらの環境変数がいずれも定義されていない場合、テンポラリファイルは、Windows の場合は現在のディレクトリ、UNIX の場合は */tmp* ディレクトリに作成されます。

Windows Mobile では、*TempFolder* レジストリキーエントリによってテンポラリファイルのディレクトリが決定されます。これが定義されていない場合は、*¥Temp* が使用されます。このディレクトリが存在せず、作成できない場合は、現在のディレクトリが使用されます。「Windows Mobile でのレジストリ設定」466 ページを参照してください。

- **通信関連のテンポラリファイル (クライアントとデータベースサーバーの両方に対して UNIX のみで作成)**　UNIX では、通信関連のテンポラリファイルは -dt で指定されたディレクトリには保存されません。その代わりに、データベースサーバーは以下の環境変数をこのままの順序で調べ、テンポラリファイルの保存先にするディレクトリを決定します。

| Unix |
| --- |
| 1. SATMP |
| 2. TMP |
| 3. TMPDIR |
| 4. TEMP |

これらの環境変数がいずれも定義されていない場合、テンポラリファイルは UNIX では */tmp* ディレクトリに作成されます。

UNIX では、共有メモリを介して接続する場合、クライアントとデータベースサーバーの両方で SATMP を同じ値に設定する必要があります。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照


- 「データベースファイルのタイプ」112 ページ
- 「SATMP 環境変数」451 ページ
- 「異なるファイルの異なるデバイスへの配置」『SQL Anywhere サーバー SQL の使用法』
- 「sa_disk_free_space システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「temp_space_limit_check オプション」695 ページ
- 「テンポラリファイル用の十分なディスク領域があるかを確認する」1211 ページ


### 例

データベースサーバー関連のテンポラリファイルの場所を特定するには、DB_PROPERTY システム関数で TempFileName プロパティを使用します。

```
SELECT DB_PROPERTY ( 'TempFileName' );
```

## -ec dbeng12/dbsrv12 サーバーオプション

トランスポートレイヤーセキュリティまたは暗号化を使用して、すべてのクライアントとの間で転送されるすべての Command Sequence 通信プロトコルパケット (DBLib、ODBC、OLE DB) を暗号化します。TDS パケットは暗号化されません。

### 構文

**{ dbeng12 | dbsrv12 } -ec** *encryption-options* ...

*encryption-options* :

**{ NONE |**
**SIMPLE |**
**TLS ( TLS_TYPE=***algorithm***;**
**[ FIPS={ Y | N }; ]**
**IDENTITY=***server-identity-filename***;**
**IDENTITY_PASSWORD=***password* **) }**, ...

### 指定可能な値

- **NONE**　暗号化されない接続を受け入れます。

- **SIMPLE**　単純暗号化された接続を受け入れます。このタイプの暗号化は、すべてのプラットフォームで、また以前のバージョンのデータベースサーバーとクライアントでサポートされます。単純暗号化では、サーバー認証、強力な楕円曲線暗号化、RSA 暗号化、その他のトランスポートレイヤーセキュリティ機能は提供されません。

- **TLS**　暗号化される接続を受け入れます。TLS パラメーターに指定できる必須の引数は次のとおりです。

  - ***algorithm***　RSA 暗号化の場合は **RSA** を指定し、ECC 暗号化の場合は **ECC** を指定します。FIPS 認定の RSA 暗号化の場合は、**TLS_TYPE=RSA;FIPS=Y** を指定します。RSA FIPS 認定の暗号化は別の認定ライブラリを使用しますが、9.0.2 以降で RSA を指定しているクライアントと互換性があります。

FIPS 認定コンポーネントがサポートされているプラットフォームのリストについては、http://www.ianywhere.jp/tech/1061806-os_components.html を参照してください。

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

FIPS 認定のアルゴリズムの実行については、「-fips dbeng12/dbsrv12 サーバーオプション」230 ページを参照してください。

> **注意**
> バージョン 10 以降のクライアントでは、ECC アルゴリズムを使用してバージョン 9.0.2 以前のデータベースサーバーに接続することはできません。この構成で強力な暗号化が必要な場合は、RSA アルゴリズムを使用してください。

○ ***server-identity-filename*** サーバー ID 証明書のパスとファイル名を指定します。FIPS 認定の RSA 暗号化を使用している場合は、RSA アルゴリズムを使用して証明書を生成する必要があります。

サーバー証明書の作成については、「デジタル証明書」1414 ページを参照してください。サーバー証明書は、自己署名証明書、または認証局やエンタープライズルート証明書の署名を受けた証明書のいずれかです。

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

### 適用対象

NONE と SIMPLE は、すべてのサーバーとオペレーティングシステムに適用されます。

TLS は、すべてのサーバーとオペレーティングシステムに適用されます (Windows Mobile を除く)。

FIPS 認定の暗号化サポートの詳細については、http://www.ianywhere.jp/tech/1061806-os_components.html を参照してください。

### 備考

このオプションは、トランスポートレイヤーセキュリティを使用してクライアントアプリケーションとデータベースサーバー間の通信パケットを安全化する場合に使用します。「トランスポートレイヤーセキュリティ」1410 ページを参照してください。

> **注意**
> 別途ライセンスが必要な必須コンポーネント
>
> ECC 暗号化と FIPS 認定の暗号化には、別途ライセンスが必要です。強力な暗号化テクノロジはすべて、輸出規制対象品目です。
>
> 「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

-ec オプションを指定すると、データベースサーバーは指定された暗号化タイプによって暗号化される接続のみ受け入れます。カンマ区切りリストで、少なくとも 1 つのサポートされているパ

ラメーターを指定してください。TDS プロトコルを介した接続は、jConnect を使用する Java アプリケーションを含みますが、-ec オプションの使用に関係なく常に受け入れられ、暗号化されることはありません。この TDS プロトコルオプションを NO に設定すると、これらの暗号化されていない TDS 接続は禁止されます。「TDS プロトコルオプション (サーバー側のみ)」399 ページを参照してください。

デフォルトでは、通信パケットは暗号化されないため、セキュリティに潜在的なリスクがあります。ネットワークパケットのセキュリティが心配な場合は、-ec オプションを使用します。暗号化がパフォーマンスに及ぼす影響はごくわずかです。

データベースサーバーが単純暗号化を受け入れ、暗号化されない接続を受け入れない場合、暗号化を使用しない TDS 接続以外の接続では、単純暗号化が使用されます。

-ec SIMPLE を指定してデータベースサーバーを起動すると、データベースサーバーは単純暗号化を使用した接続だけを受け入れます。TLS 接続 (ECC、RSA、RSA FIPS 認定の暗号化) は失敗し、暗号化を要求しない接続では単純暗号化が使用されます。

-ec SIMPLE,TLS(TLS_TYPE=ECC) を指定してデータベースサーバーを起動すると、データベースサーバーは ECC 暗号化または単純暗号化を使用する接続だけを受け入れます。RSA または RSA FIP 認定の暗号化を使用した接続は失敗し、暗号化を要求しない接続では単純暗号化が使用されます。

データベースサーバーで TCP/IP 上の暗号化された接続を受け入れ、さらに共有メモリを介してローカルコンピューターのデータベースへも接続できるようにする場合は、データベースサーバーの起動時に -ec オプションとともに -es オプションを指定できます。「-es dbeng12/dbsrv12 サーバーオプション」227 ページを参照してください。

*dbecc12.dll* ファイルと *dbrsa12.dll* ファイルには、暗号化と復号化に使用される ECC コードと RSA コードが保存されています。*dbfips12.dll* ファイルには、FIPS 認定の RSA アルゴリズムのコードが含まれています。データベースサーバーに接続するときに、適切なファイルが見つからなかったり、エラーが発生したりすると、データベースサーバーメッセージウィンドウにメッセージが表示されます。指定されたタイプの暗号化を開始できない場合、サーバーは起動しません。

クライアントとサーバーで暗号化の設定が一致していることが必要です。設定が異なっていると、次の場合を除き、接続は失敗します。

- データベースサーバーに対して -ec SIMPLE を指定し、-ec NONE を指定しなかった場合、暗号化を要求しない接続は許可され、自動的に単純暗号化が使用されます。

- データベースサーバー側で RSA を指定し、クライアント側で FIPS 認定の暗号化を指定している場合、またはその逆の場合には、接続は成功します。この場合、Encryption 接続プロパティはデータベースサーバー側で指定された値を返します。

### 参照


- 「トランスポートレイヤーセキュリティを使用するデータベースサーバー」1420 ページ
- 「Encryption (ENC) 接続パラメーター」342 ページ
- 「-ek dbeng12/dbsrv12 データベースオプション」310 ページ
- 「-ep dbeng12/dbsrv12 サーバーオプション」226 ページ
- 「-es dbeng12/dbsrv12 サーバーオプション」227 ページ
- 「DatabaseKey (DBKEY) 接続パラメーター」335 ページ
- 「FIPS プロトコルオプション」385 ページ


### 例

次の例は、暗号化されない接続と単純暗号化を使用する接続を許可します。

```
dbsrv12 -ec NONE,SIMPLE -x tcpip c:¥mydemo.db
```

次の例は、楕円曲線サーバー証明書 *eccserver.id* を使用するデータベースサーバーを起動します。

```
dbsrv12 -ec TLS(TLS_TYPE=ECC;IDENTITY=eccserver.id;IDENTITY_PASSWORD=test) -x tcpip c:
¥mydemo.db
```

次の例は、RSA サーバー証明書 *rsaserver.id* を使用するデータベースサーバーを起動します。

```
dbsrv12 -ec TLS(TLS_TYPE=RSA;IDENTITY=rsaserver.id;IDENTITY_PASSWORD=test) -x tcpip c:
¥mydemo.db
```

次の例は、FIPS 認定の RSA サーバー証明書 *rsaserver.id* を使用するデータベースサーバーを起動します。

```
dbsrv12 -ec TLS(TLS_TYPE=RSA;FIPS=Y;IDENTITY=rsaserver.id;IDENTITY_PASSWORD=test) -x
tcpip c:¥mydemo.db
```

## -ep dbeng12/dbsrv12 サーバーオプション

強力に暗号化されたデータベースを起動するときに、暗号化キーをユーザーに要求します。

### 構文

**{ dbeng12 | dbsrv12 } -ep** ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

-ep オプションを指定すると、データベースサーバーは、コマンドラインから起動されるときに暗号化キーの入力が必要なデータベースについて、ユーザーが暗号化キーを入力するためのウィンドウを表示します。このサーバーオプションでは、クリアテキストで暗号化キーを見ることができないようにすることで、高いセキュリティが得られます。

このオプションを指定した場合、次の条件がすべて該当する場合にユーザーは暗号化キーの入力を要求されます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

- -ep オプションが指定されている
- サーバーが Windows パーソナルサーバーであるか、サーバーが起動したばかりである
- データベースを起動するキーが必要である
- データベースサーバーが Windows サービスではない、またはデスクトップとの対話オプションがオンになった Windows サービスである
- サーバーがデーモンではない (UNIX)。

クライアントアプリケーションとデータベースサーバー間の通信パケットを安全化するには、-ec サーバーオプションとトランスポートレイヤーセキュリティを使用します。「トランスポートレイヤーセキュリティ」1410 ページを参照してください。

### 参照


- 「トランスポートレイヤーセキュリティを使用するデータベースサーバー」1420 ページ
- 「-ec dbeng12/dbsrv12 サーバーオプション」223 ページ
- 「-ek dbeng12/dbsrv12 データベースオプション」310 ページ
- 「Encryption (ENC) 接続パラメーター」342 ページ
- 「DatabaseKey (DBKEY) 接続パラメーター」335 ページ


### 例

*myencrypted.db* データベースが起動すると、ユーザーは暗号化キーの入力を要求されます。

```
dbsrv12 -ep -x tcpip myencrypted.db
```

## -es dbeng12/dbsrv12 サーバーオプション

共有メモリを経由した暗号化されていない接続を許可します。

### 構文

{ **dbeng12** | **dbsrv12** } **-ec** *encryption-options* **-es** ...

### 適用対象

すべてのサーバーとオペレーティングシステム (Windows Mobile を除く)

### 備考

このオプションは、-ec オプションとともに指定された場合のみ有効です。-es オプションは、共有メモリを経由した、暗号化されていない接続を許可するようにデータベースサーバーに指定します。TCP/IP を介した接続では、-ec オプションで指定された暗号化タイプを使用する必要があります。このオプションは、リモートクライアントからのデータベースアクセスには暗号化された接続を使用し、パフォーマンス上の理由から、ローカルコンピューターからのデータベースアクセスには暗号化されていない接続を使用できるようにしたい場合に便利です。

### 参照


- 「-ec dbeng12/dbsrv12 サーバーオプション」223 ページ
- 「トランスポートレイヤーセキュリティを使用するデータベースサーバー」1420 ページ


### 例

次の例は、単純暗号化を使用する接続と、共有メモリを経由した暗号化されない接続を許可します。

```
dbsrv12 -ec SIMPLE -es -x tcpip c:¥mydemo.db
```

## -f dbeng12/dbsrv12 サーバーリカバリオプション

トランザクションログが失われた後で、データベースサーバーを強制的に起動します。

### 構文

**{ dbeng12 | dbsrv12 } -f** ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

> **警告**
> このオプションは、リカバリの場合にだけ使用されます。

トランザクションログが存在しない場合、データベースサーバーはデータベースのチェックポイントリカバリを実行して終了します。データベースサーバーの実行は継続されません。この後、-f オプションを使用せずにデータベースサーバーを再起動し、通常の操作を行うことができます。

データベースと同じディレクトリにトランザクションログが存在する場合、データベースサーバーはチェックポイントリカバリとトランザクションログを使ったリカバリを実行して終了します。データベースサーバーの実行は継続されません。この後、-f オプションを使用せずにデータベースサーバーを再起動し、通常の操作を行うことができます。

サーバーの起動時にキャッシュサイズを指定すると、リカバリ時間を短縮できます。

### 参照


- 「特殊モード」160 ページ
- 「バックアップとデータリカバリ」1071 ページ


### 例

次のコマンドを入力すると、データベースサーバーがデータベース *mydatabase.db* を起動し、リカバリを実行します。

```
dbeng12 mydatabase.db -f
```

# -fc dbeng12/dbsrv12 サーバーオプション

ファイルシステムフルのコールバック関数がある DLL (UNIX では共有オブジェクト) のファイル名を指定します。

### 構文

{ **dbeng12** | **dbsrv12** } **-fc** *filename* ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

このオプションを使用すると、ファイルシステムフルの状態が発生したときに、ユーザーに通知して、必要に応じて修正措置をとることができます。-fc オプションを使用した場合、起動時にデータベースサーバーは指定の DLL の読み込みとコールバック関数のエントリポイントの解決を試みます。DLL とエントリポイントの両方を検出できない場合、データベースサーバーはエラーを返し、停止します。ユーザーが提供する DLL は、コールバックを使用して指定のバッチファイル (UNIX の場合はシェルスクリプト) を呼び出し、診断または修正措置をとることができます。また、コールバック関数自体でもそのようなアクションを実行できます。

*%SQLANYSAMP12%¥SQLAnywhere¥DiskFull* にディスクフルのコールバック関数のサンプルがあります。

データベースサーバーは、他の DLL やファイルを検索する場合と同じロケーションでコールバック関数 DLL を検索します。

データベースサーバーがファイルを検索する場所の詳細については、「SQL Anywhere のファイル検索方法」460 ページを参照してください。

データベースサーバーは、ディスクフルの状態を検出すると、コールバック関数 (指定されている場合) を呼び出し、それに以下の情報を渡します。

●条件がトリガーされた DB 領域の名前

●失敗した操作からのオペレーティングシステム固有のエラーコード

xp_out_of_disk からのリターンコードにより、原因となった操作を中止するか再実行するかが決定されます。0 以外の値が返された場合は、操作は中止され、それ以外の場合は操作が再実行されます。0 が返され、ファイルシステムの操作が失敗するかぎり、コールバック関数が繰り返し呼び出されます。

Microsoft Windows プラットフォームでは、データベースサーバーの起動時に、データベースサーバーメッセージウィンドウが表示される設定で (-qi も -qw も指定されていない)、コールバック DLL が指定されていない場合は、ディスクフルの状態が発生するとウィンドウが表示されます。このウィンドウには、DB 領域名とエラーコードが含まれているので、ユーザーはディスクフルの状態の原因となった操作を再実行するか中止するかを決定できます。

その他すべてのオペレーティングシステムでは、-fc が指定されていない場合にディスクフルの状態になると、致命的なエラーが発生します。

データベースファイル、ログファイル、テンポラリファイルなどを格納しているデバイスの使用可能なディスク領域を管理するシステムイベントを作成して、ディスク領域の不足を管理者に警告できます。

「CREATE EVENT 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

### 参照


- 「コールバック関数」『SQL Anywhere サーバー プログラミング』
- 「システムイベント」1126 ページ
- 「max_temp_space オプション」647 ページ
- 「temp_space_limit_check オプション」695 ページ


### 例

データベースサーバーは、起動時に *diskfull.dll* DLL をロードしようとします。

```
dbeng12 -fc diskfull.dll -n my_server
```

## -fips dbeng12/dbsrv12 サーバーオプション

強力なデータベースと通信の暗号化に FIPS 認定のアルゴリズムのみを使用することを要求できます。

### 構文

**{ dbeng12 | dbsrv12 } -fips** ...

### 適用対象

Windows、UNIX、Linux

### 備考

このオプションを指定すると、すべてのデータベースサーバー暗号化で FIPS 認定のアルゴリズムが使用されます。このオプションは、データベースの強力な暗号化、クライアント／サーバーのトランスポートレイヤーセキュリティ、Web サービスのトランスポートレイヤーセキュリティに適用されます。-fips オプションが指定されているときには、暗号化されていない接続とデータベースは使用できますが、単純暗号化は使用できません。

> **注意**
> 別途ライセンスが必要な必須コンポーネント
>
> ECC 暗号化と FIPS 認定の暗号化には、別途ライセンスが必要です。強力な暗号化テクノロジはすべて、輸出規制対象品目です。
>
> 「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

データベースの強力な暗号化では、-fips オプションを指定すると、CREATE DATABASE 文の ALGORITHM 句で FIPS 認定相当の AES と AES256 が指定されていても、新しいデータベースでそれが使用されます。

-fips を指定してデータベースサーバーを起動した場合、AES、AES256、AES_FIPS、AES256_FIPS の暗号化方式で暗号化されたデータベースを実行できますが、単純暗号化方式で暗号化されたデータベースは実行できません。暗号化されていないデータベースはサーバーで開始できます。

AES_FIPS または AES256_FIPS で暗号化したデータベースを実行するために使用するコンピューターには、SQL Anywhere セキュリティオプションをインストールしてください。

トランスポートレイヤーセキュリティでは、-fips オプションを指定すると、RSA が指定されていても、サーバーは FIPS 認定の RSA アルゴリズムを使用します。ECC を指定した場合、FIPS 認定の楕円曲線アルゴリズムは使用できないため、エラーが発生します。

Web サービス用のトランスポートレイヤーセキュリティでは、-fips オプションを指定すると、HTTPS が指定されていてもサーバーは FIPS 認定の HTTPS FIPS を使用します。

-fips を指定すると、ENCRYPT 関数と HASH 関数では FIPS 認定の RSA 暗号化アルゴリズムが使用され、パスワードハッシングではアルゴリズムとして SHA-256 ではなく SHA-256 FIPS が使用されます。

### 参照


- 「強力な暗号化」1397 ページ
- 「トランスポートレイヤーセキュリティ」1410 ページ
- 「SQL Anywhere Web サービスの暗号化」1425 ページ
- 「-ec dbeng12/dbsrv12 サーバーオプション」223 ページ
- 「ENCRYPT 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』
- 「HASH 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』


## -ga dbeng12/dbsrv12 サーバーオプション

最後の非 HTTP クライアント接続を切断した後、データベースをアンロードします。

### 構文

**{ dbeng12 | dbsrv12 } -ga** ...

### 適用対象

すべてのオペレーティングシステム

### 備考

ネットワークサーバーでこのオプションを指定すると、最後の非 HTTP クライアント接続が切断された後に各データベースがアンロードされます。最後の非 HTTP クライアント接続が切断された後に各データベースをアンロードし、最後のデータベースが停止したときにはサーバーも停止します。

データベースへの唯一の接続が HTTP 接続で、自動的に停止するようにデータベースが設定されている場合、HTTP 接続が切断したときにデータベースはアンロードされません。また、-ga オプションを指定したデータベースに HTTP 接続と Command Sequence 接続または TDS 接続がある場合は、最後の Command Sequence 接続または TDS 接続が切断したときにデータベースが自動的に停止します。このときに HTTP 接続がまだあった場合は切断されます。

### 参照

- 「データベースの再構築」『SQL Anywhere サーバー SQL の使用法』
- 「AutoStop (ASTOP) 接続パラメーター」324 ページ

## -gb dbeng12/dbsrv12 サーバーオプション

サーバープロセスの優先度クラスを設定します。

### Windows 構文

{ **dbeng12** | **dbsrv12** } **-gb** { **idle** | **normal** | **high** | **maximum** } ...

### UNIX/Mac OS X 構文

{ **dbeng12** | **dbsrv12** } **-gb** *level* ...

### 指定可能な値

- **UNIX**　UNIX の場合、*level* は -20 ～ 19 の整数です。UNIX のデフォルト値は、親プロセスの nice 値と同じです。*level* 値を下げると、より適切なスケジューリング優先度になります。-gb オプションには、nice 値の設定に関するすべての制限が適用されます。たとえば、ほとんどの UNIX プラットフォームの場合、プロセスの優先度レベルを下げる (たとえば、0 から -1 に変更する) ことができるのは root ユーザーだけです。

- **Windows**　Windows の場合、一般的な設定として使用されるのは、normal または high です。値 idle は万全を期すために用意されており、maximum はコンピューターの実行に影響を及ぼす可能性があります。

### 適用対象

Windows、UNIX、Mac OS X

「クラウドに関する注意点」このオプションは、クラウドのサーバーでは完全にテストされていません。

## -gc dbeng12/dbsrv12 サーバーオプション

チェックポイント間の最大間隔を設定します。

### 構文

{ **dbeng12** | **dbsrv12** } **-gc** *minutes* ...

### デフォルト

60 分

### 指定可能な値

- **minutes**　デフォルト値は、checkpoint_time データベースオプションの設定値 (デフォルトは 60 分) です。値 0 を入力すると、デフォルト値の 60 が使用されます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

このオプションは、各データベースでチェックポイントを行わずに、データベースサーバーを実行する期待最大時間を分数で設定するために使用します。

通常、チェックポイントは、指定する時間より頻繁に発生します。

「データベースサーバーがチェックポイントのタイミングを決定する方法」1114 ページを参照してください。

### 参照


- 「checkpoint_time オプション」603 ページ
- 「チェックポイントログ」133 ページ
- 「データベースサーバーがチェックポイントのタイミングを決定する方法」1114 ページ


## -gd dbeng12/dbsrv12 サーバーオプション

データベースサーバー上でデータベースを起動または停止するために必要なパーミッションを設定します。

### 構文

**{ dbeng12 | dbsrv12 } -gd { DBA | all | none } ...**

### 指定可能な値

- **DBA**　DBA 権限のあるユーザーだけがデータベースを起動または停止できます。
- **all**　すべてのユーザーがデータベースを起動または停止できます。リモートクライアントからアクセスできるネットワークサーバーへの設定はおすすめしません。
- **none**　データベースサーバー自体が起動しているか停止している場合を除き、データベースの開始と停止は許可されません。

### デフォルト

パーソナルデータベースサーバーのデフォルト設定は **all** で、ネットワークデータベースサーバーのデフォルト設定は **DBA** です。大文字と小文字の両方の構文を使用できます。

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

このオプションは、ユーザーが新しいデータベースファイルをデータベースサーバーにロードするために、または実行中のデータベースサーバーでデータベースを停止するために必要なパーミッションのレベルを指定します。

オプションを DBA に設定した場合、クライアントアプリケーションは、データベースを開始または停止させるために同一のサーバー上で動作している別のデータベースへの既存の DBA 接続を使用する必要があります。サーバーへの接続があるだけでは十分ではありません。

次のように StartDBPermission サーバープロパティを使用して -gd オプションの設定を取得できます。

```
SELECT PROPERTY ( 'StartDBPermission' );
```

データベースサーバーを停止するためのパーミッションは、-gk オプションで指定されます。

### 参照

- 「-gk dbeng12/dbsrv12 サーバーオプション」235 ページ
- 「パーミッション」520 ページ
- 「STOP DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』

### 例

ネットワークデータベースサーバーで -gd オプションを使用する手順は、次のとおりです。

1. ネットワークデータベースサーバーを起動します。

   ```
   dbsrv12 -su mypwd -gd DBA -n my_server
   ```

2. Interactive SQL からユーティリティデータベースに接続します。

   ```
   dbisql -c "UID=DBA;PWD=mypwd;DBN=utility_db"
   ```

3. データベースを起動します。

   ```
   START DATABASE 'demo.db';
   ```

4. データベースに接続します。

   ```
   CONNECT USING 'DBN=demo;UID=DBA;PWD=sql';
   ```

## -ge dbeng12/dbsrv12 サーバーオプション

外部関数のスタックサイズを設定します。

### 構文

{ **dbeng12** | **dbsrv12** } **-ge** *integer* ...

### デフォルト

32 KB

### 適用対象

Windows

「クラウドに関する注意点」このオプションは、クラウドのサーバーでは完全にテストされていません。

### 備考

外部関数を実行するスレッドのスタックサイズをバイト数で設定します。

### 参照

- 「スレッド動作」164 ページ

## -gf dbeng12/dbsrv12 サーバーオプション

サーバー上でトリガーを起動不可にします。

### 構文

**{ dbeng12 | dbsrv12 } -gf** ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

-gf サーバーオプションは、トリガーの起動を無効にするようにサーバーに指示します。

### 参照

- 「fire_triggers オプション」624 ページ
- 「トリガー」『SQL Anywhere サーバー SQL の使用法』

## -gk dbeng12/dbsrv12 サーバーオプション

dbstop を使用して、ネットワークサーバーとパーソナルサーバーを停止するために必要なパーミッションを設定します。

### 構文

**{ dbeng12 | dbsrv12 } -gk { DBA | all | none }** ...

### 指定可能な値

- **DBA**　DBA 権限を持つユーザーだけが dbstop を使ってデータベースサーバーを停止できます。これはネットワークサーバーのデフォルトです。
- **all**　データベースサーバーを停止するには、パーミッションは必要ありません。これはパーソナルサーバーのデフォルトです。
- **none**　dbstop を使ってデータベースサーバーを停止できません。

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 参照

● 「サーバー停止ユーティリティ (dbstop)」1024 ページ

## -gl dbeng12/dbsrv12 サーバーオプション

LOAD TABLE を使用するデータのロード、UNLOAD または UNLOAD TABLE を使用するデータのアンロードに必要なパーミッションを設定します。

### 構文

**{ dbeng12 | dbsrv12 } -gl { DBA | all | none }** ...

### 指定可能な値

- **DBA**　DBA 権限を持つユーザーだけが、データベースからデータをロード／アンロードできます。
- **all**　すべてのユーザーがデータベースからデータをロード／アンロードできます。
- **none**　データのロードやアンロードはできません。

### デフォルト

UNIX 以外のオペレーティングシステムを使用するパーソナルデータベースサーバーの場合、デフォルト設定は all です。ネットワークデータベースサーバーや UNIX パーソナルサーバーの場合、デフォルト設定は DBA です。

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

UNLOAD TABLE や UNLOAD 文は、データベースサーバーコンピューターのファイルにデータを配置します。LOAD TABLE 文は、データベースサーバーコンピューターからファイルを読み込みます。

これらの文を使用してファイルシステムへのアクセスを制御するには、-gl サーバーオプションを使用します。このオプションによって、これらの文を使用するために必要なデータベースパーミッションのレベルを制御できます。

大文字と小文字の両方の構文を使用できます。

デフォルト設定は、UNIX 以外のプラットフォームではパーソナルデータベースサーバーが現在のコンピューターで実行され、したがってユーザーはすでにファイルシステムにアクセスしているという事実を反映しています。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照

- 「LOAD TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「UNLOAD 文」『SQL Anywhere サーバー SQL リファレンス』

## -gm dbeng12/dbsrv12 サーバーオプション

データベースサーバーに対する同時接続の数を制限します。

### 構文

{ **dbeng12** | **dbsrv12** } **-gm** *integer* ...

### デフォルト

パーソナルサーバーのデフォルト値は 10 です。ネットワークデータベースサーバーのデフォルト値は 32766 ですが、この数値は、操作時にサーバーで使用される内部テンポラリ接続によって削減されます。

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

「クラウドに関する注意点」このオプションは、クラウドのサーバーでは完全にテストされていません。

### 備考

サーバーの接続制限を定義します。ライセンス契約に許可されている数より大きい値、もしくはメモリ制約を超えた値をここで設定した場合、その値は無効です。通常は、コンピューターリソースの制約から、ネットワークサーバーへの接続の最大数はデフォルト値より小さくなります。

DBA 権限を持つユーザーがデータベースサーバーに接続して他の接続を削除できるように、データサーバーは、接続制限より 1 つ余分に DBA 接続を許可します。

### 参照

- MaxConnections サーバープロパティ764 ページ

## -gn dbsrv12 サーバーオプション

データベースサーバーのマルチプログラミングレベルを設定します。

### 構文

**dbsrv12 -gn** *integer* ...

### デフォルト

ネットワークデータベースサーバーとパーソナルデータベースサーバーのどちらの場合も 20 のアクティブタスク

Windows Mobile では 3

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

-gn サーバーオプションは、データベースサーバーの初期マルチプログラミングレベルを設定します。

デフォルトでは、サーバーの負荷に応じて、マルチプログラミングレベルがネットワークデータベースサーバーで自動的に調整されます。ネットワークデータベースサーバーの起動後に、sa_server_option システムプロシージャーと CurrentMultiprogrammingLevel パラメーターを使用して、マルチプログラミングの設定を手動で上書きできます。

**注意**
大きな*整数*値を設定することによって、32 ビットプラットフォームで利用可能な最大キャッシュサイズを減らすことができます。

### 参照


- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「SQL Anywhere のスレッド化」161 ページ
- 「マルチプログラミングレベルのデータベースサーバー設定」165 ページ
- 「-gna dbsrv12 サーバーオプション」238 ページ
- 「-gnh dbsrv12 サーバーオプション」239 ページ
- 「-gnl dbsrv12 サーバーオプション」240 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』


## -gna dbsrv12 サーバーオプション

ネットワークデータベースサーバーのマルチプログラミングレベルの自動チューニングを制御します。

### 構文

**dbsrv12 -gna** { **0** | **1** }

### デフォルト

1 (有効)

### 適用対象

すべてのオペレーティングシステム (Windows Mobile を除く)

ネットワークデータベースサーバー (dbsrv12) のみ。

### 備考

このオプションは、ネットワークデータベースサーバーのマルチプログラミングレベルの動的チューニングを有効または無効にします。sa_server_option システムプロシージャーを使用して、ネットワークデータベースサーバーの稼働中にマルチプログラミングレベルを手動で変更できます。

デフォルトでは、ネットワークデータベースサーバーのマルチプログラミングレベルの自動チューニングは有効になっています。

### 参照


- 「SQL Anywhere のスレッド化」161 ページ
- 「マルチプログラミングレベルのデータベースサーバー設定」165 ページ
- AutoMultiProgrammingLevel サーバープロパティ754 ページ
- 「max_query_tasks オプション」644 ページ
- 「-gn dbsrv12 サーバーオプション」237 ページ
- 「-gnh dbsrv12 サーバーオプション」239 ページ
- 「-gnl dbsrv12 サーバーオプション」240 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』


## -gnh dbsrv12 サーバーオプション

最大マルチプログラミングレベル (ネットワークデータベースサーバーが同時に実行できるタスクの最大数) を設定します。

### 構文

**dbsrv12 -gnh** *integer* ...

### デフォルト

-gn データベースサーバーオプション値の 4 倍

### 適用対象

すべてのオペレーティングシステム (Windows Mobile を除く)

ネットワークデータベースサーバー (dbsrv12) のみ。

### 備考

このオプションは、ネットワークデータベースサーバーの最大マルチプログラミングレベルを設定します。ネットワークデータベースサーバーが同時に実行できるタスク (ユーザー要求とシステム要求) の最大数を制限します。ネットワークデータベースサーバーが最大数を超える要求を受け取った場合、新しい要求は実行中のタスクが完了するまで待つことになります。同時に実行できるアクティブタスクの数は、ネットワークデータベースサーバーのスレッドの数と、使用されている論理プロセッサーの数によって異なります。

未スケジュールの要求とアクティブな要求を加算した総数は、-gm ネットワークデータベースサーバーオプションで制限されます。これにより、ネットワークデータベースサーバーへの接続数を制限します。

-gnh 値の設定が大きすぎると、システムのアドレス領域の大部分がスタック領域に消費されるため、エラーになる場合があります。

ネットワークデータベースサーバーのカーネルは、スケジューリングの単位としてタスクを使用します。ユーザー要求を実行するには、最低 1 つのタスクが必要です。ただし、1 つの要求に伴って追加のタスクがスケジューリングされることがあります。たとえば、外部のプロシージャーや関数 (Java、Perl、CLR など) の実行を伴う要求では、ネットワークデータベースサーバーに対してデータベース要求が発行されます。

クエリ内並列処理が関連する場合は、同時に実行される各アクセスプランコンポーネントがタスクになります。これらのタスクは、別個の要求であるかのように -gnh 制限で考慮されます。ただし、クエリ内並列処理で作成されたタスクは、アクティブな要求とアクティブでない要求の数を追跡するデータベースプロパティには反映されません。

**警告**
-gss によって指定したサイズのスタックが、各データベースサーバーワーカーに割り付けられます。ワーカーの最大数は、ネットワークサーバーの場合は -gnh オプション、パーソナルサーバーの場合は -gn オプションで指定します。-gss に大きい値を設定し、かつワーカーの最大数に大きな値を設定すると、データベースサーバーが起動しないか、キャッシュサイズが大幅に制限される場合があります。たとえば、ネットワークデータベースサーバーの起動時に -gss オプションで 16 M を指定し、-gnh オプションで 100 を指定すると、スタック用に 1.6 GB のサーバーアドレス領域が割り当てられます。

### 参照


- 「SQL Anywhere のスレッド化」161 ページ
- 「マルチプログラミングレベルのデータベースサーバー設定」165 ページ
- MaxMultiProgrammingLevel サーバープロパティ764 ページ
- 「max_query_tasks オプション」644 ページ
- 「-gn dbsrv12 サーバーオプション」237 ページ
- 「-gna dbsrv12 サーバーオプション」238 ページ
- 「-gnl dbsrv12 サーバーオプション」240 ページ
- 「-gss dbeng12/dbsrv12 サーバーオプション」243 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』


## -gnl dbsrv12 サーバーオプション

ネットワークサーバーの最小マルチプログラミングレベルを設定します。

### 構文

**dbsrv12 -gnl** *integer* ...

### デフォルト

-gtc サーバーオプション値とコンピューターの論理 CPU 数の小さい方の値。

### 適用対象

すべてのオペレーティングシステム (Windows Mobile を除く)

ネットワークデータベースサーバー (dbsrv12) のみ。

### 備考

-gnl の値を設定すると、現在のマルチプログラミングレベルをこの値より低く設定できないようになります。負荷の高い操作の後にアイドル期間が続く場合は、-gnl の値を大きくすると、負荷の高い処理が再度発生したときにネットワークデータベースサーバーのパフォーマンスを向上するのに効果的です。

### 参照


- 「-gtc dbeng12/dbsrv12 サーバーオプション」245 ページ
- 「SQL Anywhere のスレッド化」161 ページ
- 「マルチプログラミングレベルのデータベースサーバー設定」165 ページ
- MinMultiProgrammingLevel サーバープロパティ765 ページ
- 「max_query_tasks オプション」644 ページ
- 「-gn dbsrv12 サーバーオプション」237 ページ
- 「-gna dbsrv12 サーバーオプション」238 ページ
- 「-gnh dbsrv12 サーバーオプション」239 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』


## -gns dbsrv12 サーバーオプション

データベースサーバーメッセージログにマルチプログラミングレベルの自動調整の統計を表示します。

### 構文

**dbsrv12 -gns** ...

### デフォルト

統計を表示しません。

### 適用対象

すべてのオペレーティングシステム (Windows Mobile を除く)

ネットワークデータベースサーバーのみ。

### 備考

マルチプログラミングレベルの統計は、-gns オプションが指定されている場合にのみデータベースサーバーメッセージログに記録されます。

データベースサーバーの起動後に、sa_server_option システムプロシージャーを使用して、マルチプログラミングレベルの統計のレポート設定を変更できます。

マルチプログラミングレベルの統計が現在レポートされているかどうかを確認するには、次のクエリを実行します。

```
SELECT PROPERTY ( 'AutoMultiProgrammingLevelStatistics' );
```

### 参照

- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「マルチプログラミングレベルのデータベースサーバー設定」165 ページ
- AutoMultiProgrammingLevelStatistics サーバープロパティ 754 ページ
- 「-gn dbsrv12 サーバーオプション」237 ページ
- 「-gna dbsrv12 サーバーオプション」238 ページ
- 「-gnh dbsrv12 サーバーオプション」239 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「データベースサーバー動作のロギング」153 ページ

## -gp dbeng12/dbsrv12 サーバーオプション

許可される最大データベースページサイズを設定します。

### 構文

**{ dbeng12 | dbsrv12 } -gp { 2048 | 4096 | 8192 | 16384 | 32768 }** ...

### デフォルト

4096 (データベースをロードせずにデータベースサーバーを起動した場合)

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

サーバーのページサイズよりも大きいページサイズのデータベースファイルはロードできません。このオプションは、サーバーのページサイズをバイト数で明示的に設定します。

デフォルトでは、サーバーページサイズは、コマンドラインで指定されたデータベースの最大ページサイズと同じ大きさです。

いずれのプラットフォームでも、このオプションを指定せずにデータベースがロードされていない状態でサーバーを起動した場合、デフォルト値は 4096 になります。

### 参照

- 「最大ページサイズの考慮事項」150 ページ
- 「適切なページサイズの使用」『SQL Anywhere サーバー SQL の使用法』

## -gr dbeng12/dbsrv12 サーバーオプション

システム障害からのリカバリに要する最長時間を分数で設定します。

### 構文

**{ dbeng12 | dbsrv12 } -gr** *minutes* ...

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### デフォルト

recovery_time データベースオプションの設定 (デフォルトでは 2 分)

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

データベースサーバーが複数のデータベースで実行されている場合、このオプションで書き換えないかぎり、最初に起動されたデータベースで指定されているリカバリ時間が使用されます。

-gr オプションで指定した値は、データベースサーバーにチェックポイントを実行する頻度を指示します。たとえば、-gr を 5 に設定した場合、データベースサーバーは頻繁にチェックポイントを実行して、リカバリが 5 分以上かからないようにします。ただし、リカバリが必要な場合は、-gr で指定した時間より長くかかったとしても、完了するまで実行されます。

リカバリ時間には、データベースのリカバリ予想時間とチェックポイント予想時間が含まれます。

### 参照


- 「recovery_time オプション」669 ページ
- 「データベースサーバーがチェックポイントのタイミングを決定する方法」1114 ページ


## -gss dbeng12/dbsrv12 サーバーオプション

データベースサーバーでワーカーごとのスタックサイズを設定します。

### 構文

{ **dbeng12** | **dbsrv12** } **-gss** { *integer*[ **k** | **m** ] } ...

### 適用対象

すべてのオペレーティングシステムとサーバー。Windows の場合、このオプションは Windows XP 以降でサポートされます。

### 備考

-gss オプションによって、メモリが限られている環境で、データベースサーバーのメモリ使用量を節約できます。

*size* には、使用するメモリ容量をバイト単位で指定します。単位をキロバイトまたはメガバイトで指定するには、それぞれ **k**、**m** を使用します。

**警告**
-gss によって指定したサイズのスタックが、各データベースサーバーワーカーに割り付けられます。ワーカーの最大数は、ネットワークサーバーの場合は -gnh オプション、パーソナルサーバーの場合は -gn オプションで指定します。-gss に大きい値を設定し、かつワーカーの最大数に大きな値を設定すると、データベースサーバーが起動しないか、キャッシュサイズが大幅に制限される場合があります。たとえば、ネットワークデータベースサーバーの起動時に -gss オプションで 16 M を指定し、-gnh オプションで 100 を指定すると、スタック用に 1.6 GB のサーバーアドレス領域が割り当てられます。

ワーカーごとのスタックサイズの最小値、最大値、デフォルトは次のとおりです。

| プラットフォーム | 最小値 | デフォルト | 最大値 |
|---|---|---|---|
| UNIX (32 ビット) | 512 KB | 512 KB | 4 MB |
| UNIX (64 ビット) | 1 MB | 1 MB | 8 MB |
| Windows (32 ビット) | 64 KB | 1 MB | 16 MB |
| Windows (64 ビット) | 64 KB | 4 MB | 256 MB |
| Windows Mobile | 64 KB | 96 KB | 512 KB |

**参照**


- 「SQL Anywhere のスレッド化」161 ページ
- 「-gss dbeng12/dbsrv12 サーバーオプション」243 ページ


## -gt dbeng12/dbsrv12 サーバーオプション

使用できる物理プロセッサーの最大数を設定します (ライセンスされたプロセッサーの数を上限とする)。このオプションは、マルチプロセッサーシステムでのみ役立ちます。

**構文**

{ **dbeng12** | **dbsrv12** } **-gt** *num-processors* ...

**指定可能な値**

- **num-processors**　1 と、次に示す項目の最小値との間の整数で値を設定できます。
  - コンピューター上の物理プロセッサーの数
  - サーバーがライセンスを受けている CPU の最大数 (CPU がライセンスされている場合)

-gt オプションに範囲外の値を指定すると、下限または上限の値が設定されます。パーソナルデータベースサーバー (dbeng12) では、サーバーは -gt オプションの値として 1 を使用します。

**適用対象**

Windows (Windows Mobile を除く)、Linux、Solaris


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 備考

パーソナルデータベースサーバーは常に単一のプロセッサーに限定されています。per-seat ライセンスの場合、ネットワークデータベースサーバーはコンピューターで使用可能なすべての CPU を使用します (デフォルト)。CPU ベースのライセンスの場合、ネットワークデータベースサーバーはライセンスを受けたプロセッサー数のみ使用可能です。ネットワークデータベースサーバーで使用できる CPU 数には、SQL Anywhere のエディションも影響する場合があります。「エディションとライセンス」『SQL Anywhere 12 紹介』を参照してください。

-gt オプションの値を指定すると、データベースサーバーは関係マスクを調整し (ハードウェアプラットフォームでサポートされている場合)、指定した物理プロセッサーの数のみを使用して実行するようにデータベースサーバーを制限します。データベースサーバーが *n* 個のプロセッサーのライセンスを受けている場合、デフォルトでは、サーバーは *n* 個の物理プロセッサーにおいて、すべての論理プロセッサー (ハイパースレッドとコア) 上で実行します。-gtc オプションを使用すると、この動作にさらに制限を加えることができます。

### 参照


- 「-gnh dbsrv12 サーバーオプション」239 ページ
- 「-gtc dbeng12/dbsrv12 サーバーオプション」245 ページ
- 「SQL Anywhere のスレッド化」161 ページ


## -gtc dbeng12/dbsrv12 サーバーオプション

データベースサーバーが許容するプロセッサー同時実行性の最大値を指定します。

### 構文

{ **dbeng12** | **dbsrv12** } **-gtc** *logical-processors-to-use* ...

### 適用対象

Linux、Solaris、Windows オペレーティングシステム (Intel 互換の x86 と x64 プラットフォームを実行するシステム、Windows Mobile を除く)

### 備考

データベースサーバーを起動すると、検出された物理プロセッサーと論理プロセッサーの数がデータベースサーバーメッセージウィンドウに表示されます。

物理プロセッサーは、コンピューターの CPU であり、「パッケージ」または「ダイ」と呼ばれることもあります。物理プロセッサーがハイパースレッドをサポートする場合や、「マルチプロセッサー」 (一般に「マルチコアプロセッサー」と呼ばれます) として設定されている場合は、追加の論理プロセッサーが存在します。オペレーティングシステムは、論理プロセッサー上でスレッドをスケジュールします。

-gtc オプションでは、データベースサーバーが使用する論理プロセッサーの数を指定できます。このオプションを指定することで、サーバーの起動時に作成されるデータベースサーバースレッドの数を制限します。これは、データベースサーバーで同時に実行できるアクティブタスクの数を制限することになります。デフォルトでは、作成されるスレッドの数は (1 + ライセンスを受けているすべての物理プロセッサー上にある論理プロセッサーの数) です。

デフォルトでは、ライセンスされた各物理プロセッサーにおいて、すべての論理プロセッサー (コアまたはハイパースレッド) の同時使用が可能です。たとえば、1 つの CPU を搭載し、ハイパースレッドをサポートするシステムについて考えてみます。データベースサーバーは、デフォルトで、1 つの物理プロセッサー上で同時に実行する 2 つのスレッドを許可します。-gtc オプションが指定されている場合で、使用される論理プロセッサーの数が、ライセンスされている物理プロセッサーで利用できる総数よりも少なくなると、データベースサーバーはラウンドロビン方式に基づいて論理プロセッサーを割り付けます。-gtc オプションに対して暗黙的に 1 を指定すると、クエリ内並列処理 (クエリの並列処理) は無効になります。クエリ内並列処理は、max_query_tasks オプションを使用して、明示的に制限したり、無条件に無効にしたりすることもできます。

### 参照


- 「max_query_tasks オプション」644 ページ
- 「-gnh dbsrv12 サーバーオプション」239 ページ
- 「-gt dbeng12/dbsrv12 サーバーオプション」244 ページ
- 「SQL Anywhere のスレッド化」161 ページ


### 例

Windows ベースの SMP コンピューターを例に挙げて考えてみます。ここでは、システムが 4 プロセッサー構成で、各物理プロセッサーにコアが 2 つある (合計 8 つの論理プロセッサーがある) ことを前提としています。物理プロセッサーは文字で、論理プロセッサー (この例では、コア) は数字でそれぞれ区別します。このシステムには 4 プロセッサーユニットとして A0、A1、B0、B1、C0、C1、D0、D1 が存在することになります。

| シナリオ | ネットワークデータベースサーバーの設定 |
|---|---|
| シングル CPU ライセンス、または -gt に 1 が指定されている | ● -gt 1<br>● -gtc 2<br>● -gnh 20<br><br>スレッドは A0 または A1 で実行できます。 |
| CPU のライセンスに制限がなく、-gtc に 5 が指定されている | ● -gt 4<br>● -gtc 5<br>● -gnh 20<br><br>スレッドは A0、A1、B0、C0、D0 で実行できます。 |
| データベースサーバーに 3 CPU ライセンスがあり、-gtc に 5 が指定されている | ● -gt 3<br>● -gtc 5<br>● -gnh 20<br><br>スレッドは A0、A1、B0、B1、C0 で実行できます。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| シナリオ | ネットワークデータベースサーバーの設定 |
|---|---|
| CPU のライセンスに制限がなく、-gtc に 1 が指定されている | ● -gt 4<br>● -gtc 1<br>● -gnh 20<br><br>スレッドは A0 でのみ実行できます。 |

## -gu dbeng12/dbsrv12 サーバーオプション

ユーティリティコマンドのパーミッションレベルを設定します。

### 構文

**{ dbeng12 | dbsrv12 } -gu { all | none | DBA | utility_db }** ...

### デフォルト

DBA

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

CREATE DATABASE や DROP DATABASE などのユーティリティコマンドのパーミッションレベルを設定します。

utility_db レベルでは、これらのコマンドをユーティリティデータベースに接続できるユーザーだけが使用できます。

この省略可能な **all** は推奨されなくなりました。

### 参照

● 「ファイル管理文の実行に必要なパーミッション」 561 ページ

## -im dbeng12/dbsrv12 サーバーオプション

データベースサーバーをインメモリで実行し、ディスクへの書き込みをなくしたり軽減したりします。

### 構文

**{ dbeng12 | dbsrv12 } -im { c | nw }** ...

### 指定可能な値

● **チェックポイント専用 (-im c)**　チェックポイント専用モードで実行すると、データベースサーバーでトランザクションログが使用されないので、コミットした最新のトランザクショ

ンにはリカバリできません。ただし、チェックポイントログは有効になっているため、データベースを最新のチェックポイントにリカバリすることができます。通常、トランザクションログを使用せずにデータベースを実行しても、データベースサーバーはコミット操作ごとにチェックポイントの設定を実行するため、パフォーマンスに影響します。ただし、データベースサーバーをチェックポイント専用モードで実行すると、データベースサーバーはコミット操作ごとにチェックポイントの設定を実行しません。

このモードは、パフォーマンスの向上が必要で、最新のチェックポイントの実行後にコミットされたトランザクションが失われても構わないアプリケーションでの使用に適しています。

チェックポイント専用モードでの実行時には、次の制限が適用されます。

1. トランザクションログは生成されない。
2. テンポラリファイルは生成されない。
3. チェックポイントは要求に応じて設定可能で、さらにデータベースサーバーの標準のチェックポイント頻度で設定される。
4. ダーティページはチェックポイントでのみディスクにフラッシュされる。

- **非書き込み (-im nw)**　非書き込みモードで実行すると、コミットされたトランザクションはディスク上のデータベースファイルに書き込まれません。データベースが停止またはクラッシュするとすべての変更は失われるため、データベースファイルは常に元の状態に保たれます。DB 領域の拡張または新規作成の要求は許可されますが、変更内容はデータベースファイルに反映されません。新しい DB 領域を作成して使用できますが、これらはディスクに書き込まれません。非書き込みモードでバックアップを作成しても、システム DB 領域の変更内容がファイルに書き込まれないので、意味がありません。

  非書き込みモードでの実行時には、次の制限が適用されます。

  1. トランザクションログは生成されない。
  2. チェックポイントログは生成されない。
  3. テンポラリファイルは生成されない。
  4. ダーティデータベースページはディスクにフラッシュされない。
  5. 元のデータベースファイルは変更されない。

**適用対象**

すべてのオペレーティングシステムとデータベースサーバー

> **注意**
> インメモリモードには別途ライセンスが必要です。「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

### 備考

この機能は、キャッシュ内のすべてのデータベースファイルを保持するのに十分な大量のメモリを使用できるシステムで使用する場合に最も効果的です。

非書き込みモードの場合 (-im nw)、元のデータベースファイルに変更内容が書き込まれないため、現在のデータベースの永続コピーが必要な場合は、dbunload ユーティリティまたは UNLOAD TABLE 文を使用してください。SQL クエリを使用して変更内容を取得することもできますが、その場合は、手動で変更内容をデータベースファイルに書き込む必要があります。

インメモリモードによるパフォーマンス向上は、アプリケーションの負荷と I/O サブシステムの処理速度によって異なります。パフォーマンスの大幅な向上は、大量データの挿入や更新を行うアプリケーションや、コミットとチェックポイントを頻繁に実行するアプリケーションで顕著に見られます。

多くの場合、インメモリモードでのパフォーマンスは、トランザクションにグローバルテンポラリテーブルを使用した場合のパフォーマンスと同等か、それ以上に改善されます。データベースへの問い合わせが大部分を占めるアプリケーションの場合は、パフォーマンスの向上がほとんど期待できません。一般に、インメモリモードの使用によるパフォーマンスの向上は、予想されるデータベースファイル全体を保持するのに十分な容量がキャッシュに事前に割り当てられている場合に最も顕著に現れます。こうすることで、アプリケーションの実行中にキャッシュを逐次増分させることによって生じるオーバーヘッドを軽減できます。

**警告**
非書き込みモードの場合、ページがキャッシュからフラッシュされないため、データベース内のデータ量が増えすぎると、使用できるキャッシュを使い果たしてしまう場合があります。このような場合、データベースサーバーはエラーを発行して、要求の処理を停止します。このような理由から、非書き込みモードの使用には注意が必要です。また、使用する場合は、アプリケーションによる使用が予想されるページの作業セット全体を保持するのに十分なキャッシュを常に確保しておく必要があります。「チェックポイント専用」モードではチェックポイントが引き続き発生するため、「非書き込み」モードと比較して、サーバーでキャッシュを使い果たしてしまうリスクは少なくなります。

LOAD TABLE 文と一部の ALTER TABLE 文では、障害の影響を部分的に取り消すため、またはエラーからのリカバリのためにチェックポイントログが使用されます。非書き込みモードでは、チェックポイントログが作成されず、一部の文が失敗するか、エラーが発生しても、文の影響を部分的に取り消すことはできません。不正または不完全なデータがテーブルに残る場合があります。「チェックポイントログ」133 ページを参照してください。

### 参照


- 「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』
- 「-c dbeng12/dbsrv12 サーバーオプション」204 ページ
- 「インメモリモードの使用」『SQL Anywhere サーバー SQL の使用法』


## -k dbeng12/dbsrv12 サーバーオプション

パフォーマンスモニター統計値の収集を制御します。

### 構文

{ **dbeng12** | **dbsrv12** } **-k** ...

### デフォルト

パフォーマンスモニター統計値が収集されます

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

データベースサーバーの起動時に -k を指定すると、パフォーマンスモニター統計値は収集されません。-k オプションは、クエリオプティマイザーが使用するカラム統計には影響しません。

このオプションは、データベースサーバーをマルチプロセッサーコンピューターで実行しており、テストによってパフォーマンスの向上が確認できた場合のみ使用してください。通常の負荷レベルでは、パフォーマンスの向上はわずかであるため、このオプションの使用は推奨できません。パフォーマンスカウンターを無効にすると、パフォーマンス上の問題が発生した場合、分析に必要な情報を取得できません。

パフォーマンスモニター統計値の収集についての設定は sa_server_option システムプロシージャーでも変更できます。

### 参照


- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- CollectStatistics サーバープロパティ 756 ページ
- 「-ks dbeng12/dbsrv12 サーバーオプション」254 ページ
- 「-ksc dbeng12/dbsrv12 サーバーオプション」255 ページ
- 「-ksd dbeng12/dbsrv12 サーバーオプション」255 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「統計をモニタリングするための Sybase Central パフォーマンスモニター機能」『SQL Anywhere サーバー SQL の使用法』


## -kl dbeng12/dbsrv12 サーバーオプション

Kerberos GSS-API ライブラリ (UNIX では共有オブジェクト) のファイル名を指定し、データベースサーバーへの Kerberos 認証接続を有効にします。

### 構文

{ **dbeng12** | **dbsrv12** } **-kl** *GSS-API-library-file* ...

### 適用対象

すべてのオペレーティングシステム (Windows Mobile を除く)

「クラウドに関する注意点」このオプションは、クラウドのサーバーでは完全にテストされていません。

### 備考

このオプションでは、Kerberos GSS-API のロケーションと名前を指定します。このオプションが必要となるのは、Kerberos クライアントでデフォルトと異なる Kerberos GSS-API ライブラリファイル名が使用されているか、データベースサーバーを実行しているコンピューターに複数の GSS-API ライブラリがインストールされている場合だけです。Kerberos クライアントのインストールと設定が完了し、データベースサーバーが SSPI を使用できない状態であることが必要です。

このオプションを指定すると、データベースサーバーに対する Kerberos 認証が有効になります。

### 参照


- 「-kr dbeng12/dbsrv12 サーバーオプション (旧式)」252 ページ
- 「-krb dbeng12/dbsrv12 サーバーオプション」253 ページ
- 「Kerberos (KRB) 接続パラメーター」350 ページ
- 「Kerberos 認証」61 ページ
- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』


### 例

次のコマンドは、Kerberos 認証に *libgssapi_krb5.so* 共有オブジェクトを使用するデータベースサーバーを起動します。

```
dbsrv12 -kl libgssapi_krb5.so -n my_server_princ /opt/myapp/kerberos.db
```

## -kp dbeng12/dbsrv12 サーバーオプション

Kerberos サーバープリンシパルを指定し、データベースサーバーへの Kerberos 認証接続を有効にします。

### 構文

{ **dbeng12** | **dbsrv12** } **-kp** *server-principal* ...

### 適用対象

すべてのオペレーティングシステム (Windows Mobile を除く)

「クラウドに関する注意点」このオプションは、クラウドのサーバーでは完全にテストされていません。

### 備考

このオプションは、データベースサーバーによって使用される Kerberos サーバープリンシパルを指定し、Kerberos 認証を有効にします。デフォルトでは、データベースサーバーが Kerberos 認証に使用するプリンシパルは *server-name@default-realm* です。*default-realm* は Kerberos クライアントに設定されたデフォルト領域です。別のサーバープリンシパルを使用する場合は、このオプションを使用します。Kerberos サーバープリンシパルの業界フォーマットは、*server-name/hostname@realm* です。

OpenClient または jConnect の Kerberos で認証された接続がデータベースサーバーに対して作成された場合、サーバープリンシパルは、アプリケーションによっても指定される必要があります。jConnect のマニュアルの SERVICE_PRINCIPAL_NAME を参照してください。

-kp オプションが指定されている場合は、-kr オプションは指定できません。

### 参照

- 「-kl dbeng12/dbsrv12 サーバーオプション」250 ページ
- 「-krb dbeng12/dbsrv12 サーバーオプション」253 ページ
- 「-kr dbeng12/dbsrv12 サーバーオプション (旧式)」252 ページ
- 「Kerberos (KRB) 接続パラメーター」350 ページ
- 「Kerberos 認証」61 ページ
- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』

### 例

次のコマンドは、Kerberos ログインを受け入れて認証にサーバープリンシパル myserver/mycomputer.example.com@EXAMPLE.COM を使用するデータベースサーバーを起動します。

```
dbeng12 -kp myserver/mycomputer.example.com@EXAMPLE.COM -n myserver C:¥kerberos.db
```

## -kr dbeng12/dbsrv12 サーバーオプション (旧式)

Kerberos サーバープリンシパルの領域を指定し、データベースサーバーへの Kerberos 認証接続を有効にします。

> **注意**
> -kr オプションを使用する方法は廃止されます。-kp オプションを使用して Kerberos サーバープリンシパルを指定することをおすすめします。-kp を指定するときに、サーバープリンシパルは、データベースサーバーを実行しているコンピューターの Kerberos keytab ファイルに抽出されている必要があります。「-kp dbeng12/dbsrv12 サーバーオプション」251 ページを参照してください。

### 構文

**{ dbeng12 | dbsrv12 } -kr** *server-realm* ...

### 適用対象

すべてのオペレーティングシステム (Windows Mobile を除く)

「クラウドに関する注意点」このオプションは、クラウドのサーバーでは完全にテストされていません。

### 備考

このオプションでは、Kerberos サーバープリンシパルの領域を指定します。通常は、データベースサーバーが Kerberos 認証に使用するプリンシパルは *server-name@default-realm* です。*default-realm* は Kerberos クライアントに設定されたデフォルト領域です。このオプションは、サーバー

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

プリンシパルの領域をデフォルト領域以外に変更する場合に使用します。このオプションを指定すると、サーバープリンシパルは *server-name@server-realm* となります。

このオプションを指定すると、データベースサーバーに対する Kerberos 認証が有効になります。

-kp オプションが指定されている場合は、-kr オプションは指定できません。

### 参照

- 「-kl dbeng12/dbsrv12 サーバーオプション」250 ページ
- 「-krb dbeng12/dbsrv12 サーバーオプション」253 ページ
- 「Kerberos (KRB) 接続パラメーター」350 ページ
- 「Kerberos 認証」61 ページ
- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』

### 例

次のコマンドは、Kerberos ログインを受け入れて認証にプリンシパル my_server_princ@MYREALM を使用するデータベースサーバーを起動します。

```
dbeng12 -kr MYREALM -n my_server_princ C:¥kerberos.db
```

## -krb dbeng12/dbsrv12 サーバーオプション

データベースサーバーへの Kerberos 認証接続を有効にします。

### 構文

**{ dbeng12 | dbsrv12 } -krb** ...

### 適用対象

すべてのオペレーティングシステム (Windows Mobile を除く)

「クラウドに関する注意点」このオプションは、クラウドのサーバーでは完全にテストされていません。

### 備考

このオプションは、データベースサーバーに対する Kerberos 認証を有効にします。データベースサーバーが Kerberos を使用してクライアントを認証するためには、-krb、-kl、-kr のうち 1 つ以上を指定する必要があります。

Kerberos 認証を使用するためには、クライアントコンピューターとデータベースサーバーコンピューターの両方に Kerberos クライアントがインストールされ、設定が完了している必要があります。さらに、Kerberos KDC にプリンシパル *server-name@REALM* があり、プリンシパル *server-name@REALM* のキータブがデータベースサーバーコンピューター上の keytab ファイルに抽出されていることも必要です。この準備が完了していない場合、-krb オプションを指定すると、データベースサーバーは起動しません。

> **注意**
> データベースサーバー名には、/、¥、@ を使用できません。また、Kerberos ではマルチバイト文字を使用したデータベースサーバー名は使用できません。

Kerberos ログインを許可するためには login_mode データベースオプションを設定する必要があります。また、GRANT KERBEROS LOGIN 文を使用して、Kerberos クライアントプリンシパルをデータベースユーザー ID にマッピングすることも必要です。

### 参照

- 「-kl dbeng12/dbsrv12 サーバーオプション」250 ページ
- 「-kr dbeng12/dbsrv12 サーバーオプション (旧式)」252 ページ
- 「Kerberos (KRB) 接続パラメーター」350 ページ
- 「Kerberos 認証」61 ページ
- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「login_mode オプション」635 ページ

### 例

次のコマンドは、データベースサーバーの Kerberos プリンシパル my_server_princ@MYREALM に対し、データベースサーバー my_server_princ を起動します。

```
dbsrv12 -krb -n my_server_princ C:¥kerberos.db
```

## -ks dbeng12/dbsrv12 サーバーオプション

データベースサーバーからカウンター値を収集するためにパフォーマンスモニターで使用される共有メモリの作成を無効にします。

### 構文

**{ dbeng12 | dbsrv12 } -ks 0** ...

### 適用対象

Windows

### 備考

このオプションを指定すると、パフォーマンスモニターには現在のデータベースサーバーのデータベースサーバー名、データベース名、接続統計は表示されません。

### 参照

- 「統計をモニタリングするための Sybase Central パフォーマンスモニター機能」『SQL Anywhere サーバー SQL の使用法』
- 「-k dbeng12/dbsrv12 サーバーオプション」249 ページ
- 「-ksc dbeng12/dbsrv12 サーバーオプション」255 ページ
- 「-ksd dbeng12/dbsrv12 サーバーオプション」255 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## -ksc dbeng12/dbsrv12 サーバーオプション

パフォーマンスモニターでモニターできる接続の最大数を指定します。

### 構文

{ **dbeng12** | **dbsrv12** } **-ksc** *integer* ...

### デフォルト

10

### 適用対象

Windows

### 参照

- 「統計をモニタリングするための Sybase Central パフォーマンスモニター機能」『SQL Anywhere サーバー SQL の使用法』
- 「-k dbeng12/dbsrv12 サーバーオプション」249 ページ
- 「-ks dbeng12/dbsrv12 サーバーオプション」254 ページ
- 「-ksd dbeng12/dbsrv12 サーバーオプション」255 ページ

## -ksd dbeng12/dbsrv12 サーバーオプション

パフォーマンスモニターでモニターできるデータベースの最大数を指定します。

### 構文

{ **dbeng12** | **dbsrv12** } **-ksd** *integer* ...

### デフォルト

2

### 適用対象

Windows

### 参照

- 「統計をモニタリングするための Sybase Central パフォーマンスモニター機能」『SQL Anywhere サーバー SQL の使用法』
- 「-k dbeng12/dbsrv12 サーバーオプション」249 ページ
- 「-ks dbeng12/dbsrv12 サーバーオプション」254 ページ
- 「-ksc dbeng12/dbsrv12 サーバーオプション」255 ページ

## -m dbeng12/dbsrv12 サーバーオプション

チェックポイントが終了すると、トランザクションログをトランケートします。

### 構文

**{ dbeng12 | dbsrv12 } -m** ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

このオプションを指定すると、シャットダウン時、またはサーバーでスケジュールされたチェックポイントの結果としてチェックポイントが実行されたときに、トランザクションログをトランケートします。

> **警告**
> このオプションを選択すると、データベースファイルを含むデバイスのメディア障害に対して無防備な状態になります。

このオプションを指定すると、トランザクションログの肥大化が自動的に制限されます。チェックポイントの頻度は、checkpoint_time と recovery_time オプションによって制御できます (また、コマンドラインでも設定できます)。

-m オプションは、高速な応答時間を必要とする大容量のトランザクションを処理する場合や、リカバリやレプリケーションがトランザクションログの内容に依存しない場合に、トランザクションログのサイズを制限するのに役立ちます。-m オプションは、各 COMMIT の後にチェックポイントが必要で、その結果パフォーマンスが低下するような場合に、トランザクションログなしで稼働する場合の代替策となります。-m オプションを指定すると、データベースファイルを含むデバイスのメディア障害に対して無防備な状態になります。-m オプションを使用する前に、トランザクションログを管理する他の代替策 (BACKUP 文やイベントの使用など) を検討してください。

データベースファイルの断片化を防ぐためには、このオプションを使用する場合に、トランザクションログをデータベースそのものとは別のデバイスまたはパーティションに保管することをおすすめします。

このオプションを指定すると、チェックポイントの実行中に他の操作は行われません。

> **警告**
> レプリケートされるデータベースまたは同期されるデータベースでは、-m オプションを使わないでください。SQL Remote と Mobile Link はトランザクションログ情報に依存します。

### 参照


- 「-m dbeng12/dbsrv12 データベースオプション」311 ページ
- 「トランザクションログ」127 ページ
- 「チェックポイントログ」133 ページ
- 「トランザクションログユーティリティ (dblog)」1041 ページ
- 「checkpoint_time オプション」603 ページ
- 「recovery_time オプション」669 ページ


 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

# -n dbeng12/dbsrv12 サーバーオプション

データベースサーバーの名前を設定します。

### 構文

{ **dbeng12** | **dbsrv12** } **-n** *server-name database-filename* ...

### デフォルト

データベースサーバーで最初に起動したのデータベースファイルの名前 (パスと拡張子を除く)。

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

データベースサーバーは起動するとき、そのコンピューターのデフォルトのデータベースサーバーになろうとします。デフォルトのサーバーがない場合、最初に起動したデータベースサーバーが、デフォルトのデータベースサーバーになります。そのコンピューターで、データベースサーバー名を明示的に指定しないで共有メモリに接続しようとすると、デフォルトのサーバーに接続されます。

> **注意**
> 配備されたアプリケーションで使用されているデータベースサーバーには -xd オプションを使用すること、またすべてのクライアントが、ServerName (Server) 接続パラメーターを使用して、接続先のデータベースサーバー名を明示的に指定することをおすすめします。このようにすると、コンピューターで複数のデータベースサーバーが実行されているときに、データベースが正しいデータベースサーバーに接続します。

サーバー名に対して文字セット変換は実行されません。クライアントとデータベースサーバーとで文字セットが異なる場合、サーバー名に拡張文字が使用されているとサーバーが見つからないことがあります。クライアントとサーバーを異なるオペレーティングシステムやロケールで実行している場合は、サーバー名に 7 ビットの ASCII 文字を使用してください。「接続文字列と文字セット」478 ページを参照してください。

データベースサーバーの名前は、有効な識別子であることが必要です。データベースサーバーのロングネームは、プロトコルごとに異なる長さにトランケートされます。データベースサーバー名には、次に該当する値を指定できません。

- 空白スペース、一重引用符、または二重引用符で始まる値
- 空白スペースで終わる値
- セミコロン、フォワードスラッシュ (/)、またはバックスラッシュ (¥) を含む値
- 長さが 250 バイトを超える値
- UNIX 上で実行するときにスペースを含む値

**注意**
Windows と UNIX では、データベースサーバーがバージョン 10.0.0 以降で、名前が次の長さを超えている場合、バージョン 9.0.2 以前のクライアントから接続することはできません。

- Windows 共有メモリの場合は、40 バイト
- UNIX 共有メモリの場合は、31 バイト
- TCP/IP の場合は、40 バイト

サーバー名は、クライアントアプリケーション接続文字列かプロファイルの ServerName (Server) 接続パラメーターで使用する名前を指定します。共有メモリ環境では、-xd を指定しないかぎり、サーバー名が指定されておらず少なくとも 1 つのデータベースサーバーがコンピューターで実行されている場合に使用されるデフォルトのデータベースサーバーがあります。

同じ名前で複数のデータベースサーバーを実行することはおすすめしません。

**注意**
2 つの -n オプションがあります。-n オプションは指定する位置によって意味が異なります。データベースファイル名の前に指定すると、サーバーオプションとしてサーバー名を指定します。データベースファイル名の後に指定すると、データベースオプションとしてデータベース名を指定します。

たとえば、次のコマンドでは、データベースサーバー名 SERV とデータベース名 DATA が指定されます。

```
dbsrv12 -n SERV sales.db -n DATA
```

「-n dbeng12/dbsrv12 データベースオプション」312 ページを参照してください。

**参照**


- 「識別子」『SQL Anywhere サーバー SQL リファレンス』
- 「ServerName (Server) 接続パラメーター」365 ページ
- 「データベースサーバー名とデータベース名を指定する」151 ページ
- 「-xd dbeng12/dbsrv12 サーバーオプション」292 ページ


**例**

*%SQLANYSAMP12%¥demo.db* ファイルでサーバーを起動するときに、-n オプションを指定しなかった場合、サーバーの名前は demo になります。

## -o dbeng12/dbsrv12 サーバーオプション

すべてのデータベースサーバーメッセージをデータベースサーバーメッセージログファイルに出力します。

**構文**

{ **dbeng12** | **dbsrv12** } **-o** *filename* ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

情報メッセージ、エラー、警告、MESSAGE 文の出力を含むすべてのデータベースサーバーメッセージを、指定したファイルとデータベースサーバーメッセージウィンドウに出力します。-o オプションとともに -qi オプションを指定すると、すべてのメッセージがデータベースサーバーメッセージログファイルだけに出力されます。

トランザクションログを使用した操作を実行するユーティリティで問題が発生する場合があるため、ファイル名の最後には *.log* を付けないようにしてください。

データベースサーバーメッセージログのファイル名を検索するには、次のコマンドを実行します。

```
SELECT PROPERTY ( 'ConsoleLogFile' );
```

### 参照


- ConsoleLogFile サーバープロパティ 757 ページ
- 「データベースサーバー動作のロギング」153 ページ
- 「-oe dbeng12/dbsrv12 サーバーオプション」259 ページ
- 「-on dbeng12/dbsrv12 サーバーオプション」260 ページ
- 「-os dbeng12/dbsrv12 サーバーオプション」261 ページ
- 「-ot dbeng12/dbsrv12 サーバーオプション」262 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「-qi dbeng12/dbsrv12 サーバーオプション」264 ページ


## -oe dbeng12/dbsrv12 サーバーオプション

起動エラー、致命的なエラー、アサーションをロギングするファイルの名前を指定します。

### 構文

**{ dbeng12 | dbsrv12 } -oe** *filename* ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

出力ログファイルの各行の先頭には、日付と時刻が記録されます。起動エラーには、次のようなエラーがあります。

●データベースファイル *database file* が開けない／読み込めない

●その名前のデータベースサーバーがすでに起動している

致命的なエラーとアサーションは、-oe が指定されているかどうかにかかわらず、Windows アプリケーションイベントログ (Window Mobile を除く) または UNIX システムログに記録されます。

トランザクションログを使用した操作を実行するユーティリティで問題が発生する場合があるため、ファイル名の最後には *.log* を付けないようにしてください。

### 参照


- 「-o dbeng12/dbsrv12 サーバーオプション」258 ページ
- 「-on dbeng12/dbsrv12 サーバーオプション」260 ページ
- 「-os dbeng12/dbsrv12 サーバーオプション」261 ページ
- 「-ot dbeng12/dbsrv12 サーバーオプション」262 ページ
- 「-qi dbeng12/dbsrv12 サーバーオプション」264 ページ


## -on dbeng12/dbsrv12 サーバーオプション

データベースサーバーメッセージログファイルの最大サイズを指定します。ログファイルがこのサイズに達すると、現在のファイルが拡張子 *.old* の付いた名前に変更され、新しいファイルが作成されます。

### 構文

**{ dbeng12 | dbsrv12 } -on {** *size*[ **k | m | g** ] **}** ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

*size* には、データベースサーバーメッセージログの最大サイズをバイト単位で指定します。単位をキロバイト、メガバイト、またはギガバイトで指定するには、それぞれ **k**、**m**、**g** のいずれかを使用してください。最小のサイズ制限は 10 KB です。デフォルトでは、最大サイズは無制限となります。

データベースサーバーメッセージログが指定されたサイズに達すると、データベースサーバーによって現在のファイルが拡張子 *.old* の付いた名前に変更され、元の名前を持つ新しいファイルが開始されます。

> **注意**
> *.old* データベースサーバーメッセージログファイルがすでに存在する場合は、そのファイルが上書きされます。古いデータベースサーバーメッセージログファイルを削除しないようにするには、代わりに -os オプションを使用します。

このオプションは、-os オプションと同時に使用できません。

トランザクションログを使用した操作を実行するユーティリティで問題が発生する場合があるため、データベースサーバーメッセージログファイル名の最後には *.log* を付けないようにしてください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照


- ConsoleLogMaxSize サーバープロパティ758 ページ
- 「データベースサーバー動作のロギング」153 ページ
- 「-o dbeng12/dbsrv12 サーバーオプション」258 ページ
- 「-oe dbeng12/dbsrv12 サーバーオプション」259 ページ
- 「-os dbeng12/dbsrv12 サーバーオプション」261 ページ
- 「-ot dbeng12/dbsrv12 サーバーオプション」262 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』


## -os dbeng12/dbsrv12 サーバーオプション

データベースサーバーメッセージログファイルの最大サイズを指定します (このサイズに達するとログのファイル名が変更されます)。

### 構文

**{ dbeng12 | dbsrv12 } -os** { *size*[ **k** | **m** | **g** ] } ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

*size* には、データベースサーバーメッセージのログを取るファイルの最大サイズをバイト単位で指定します。単位をキロバイト、メガバイト、またはギガバイトで指定するには、それぞれ **k**、**m**、**g** のいずれかを使用してください。最小のサイズ制限は 10 KB です。デフォルトでは、最大サイズは無制限となります。

データベースサーバーは、出力メッセージをデータベースサーバーメッセージログファイルに書き込む前に、現在のファイルサイズを確認します。新しいログメッセージを書き込むと、指定されたファイルサイズを超える場合は、データベースサーバーメッセージログのファイル名が *yymmddxx.slg* に変更されます。*yymmdd* は、そのファイルが作成された年、月、日を表します。*xx* は 00 で始まる番号で、1 ずつ増えていきます。

このオプションによって、データベースサーバーメッセージログファイルが古いことを確認して削除し、ディスク領域を解放できます。

このオプションは、-on オプションとは一緒に使用できません。

トランザクションログを使用した操作を実行するユーティリティで問題が発生する場合があるため、データベースサーバーメッセージログファイル名の最後には *.log* を付けないようにしてください。

### 参照


- 「データベースサーバー動作のロギング」153 ページ
- 「-o dbeng12/dbsrv12 サーバーオプション」258 ページ
- 「-oe dbeng12/dbsrv12 サーバーオプション」259 ページ
- 「-on dbeng12/dbsrv12 サーバーオプション」260 ページ
- 「-ot dbeng12/dbsrv12 サーバーオプション」262 ページ


## -ot dbeng12/dbsrv12 サーバーオプション

データベースサーバーメッセージログファイルをトランケートし、そのファイルに出力メッセージを追加します。

### 構文

{ **dbeng12** | **dbsrv12** } **-ot** *logfile* ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

このオプションの機能は、データベースサーバーメッセージログファイルのトランケート後にメッセージが書き込まれる点を除き、-o オプションと同じです。データベースサーバーメッセージログのファイル名を検索するには、次のコマンドを実行します。

```
SELECT PROPERTY ( 'ConsoleLogFile' );
```

トランザクションログを使用した操作を実行するユーティリティで問題が発生する場合があるため、データベースサーバーメッセージログファイル名の最後には *.log* を付けないようにしてください。

### 参照


- 「データベースサーバー動作のロギング」153 ページ
- 「-o dbeng12/dbsrv12 サーバーオプション」258 ページ
- 「-oe dbeng12/dbsrv12 サーバーオプション」259 ページ
- 「-on dbeng12/dbsrv12 サーバーオプション」260 ページ
- 「-os dbeng12/dbsrv12 サーバーオプション」261 ページ


## -p dbeng12/dbsrv12 サーバーオプション

通信パケットの最大サイズを設定します。

### 構文

{ **dbeng12** | **dbsrv12** } **-p** *integer* ...

### デフォルト

7300 バイト (Windows Mobile を除くすべてのオペレーティングシステム)

1460 バイト (Windows Mobile)

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

最小値は 500 バイトで、最大値は 16000 バイトです。

接続の通信バッファーサイズを変更するには、CommBufferSize (CBSIZE) 接続パラメーターを設定します。

### 参照


- 「-pc dbsrv12 サーバーオプション」263 ページ
- 「-pt dbsrv12 サーバーオプション」264 ページ
- 「CommBufferSize (CBSIZE) 接続パラメーター」326 ページ
- 「TCP/IP パフォーマンス」95 ページ


## -pc dbsrv12 サーバーオプション

同一コンピューター接続以外のすべての接続のパケットを圧縮します。

### 構文

**dbsrv12 -pc** ...

### 適用対象

すべてのオペレーティングシステムとネットワークサーバー (Web サーバーを除く)

### 備考

クライアントとサーバーの間で送信されるパケットは、-pc オプションを使用して圧縮できます。状況によっては、接続を圧縮することでパフォーマンスが向上する場合があります。大幅に圧縮可能なデータの大規模なデータ転送では、圧縮率が高くなります。クライアントの接続パラメーターに COMPRESS=NO を指定すると、特定のクライアントについてこのオプションを上書きできます。

デフォルトでは、接続は圧縮されません。-pc オプションを指定すると、同一コンピューター接続、Web サービス接続、および TDS 接続を除くすべての接続のパケットが圧縮されます。TDS 接続 (jConnect を含む) では、通信圧縮はサポートされません。

どの通信リンクを使用している場合でも、同一コンピューター接続は -pc オプションまたは COMPRESS=YES 接続パラメーターを使用しても圧縮されません。

### 参照


- 「-p dbeng12/dbsrv12 サーバーオプション」262 ページ
- 「-pt dbsrv12 サーバーオプション」264 ページ
- 「通信の圧縮設定」106 ページ
- 「Compress (COMP) 接続パラメーター」329 ページ
- 「圧縮の慎重な使用」『SQL Anywhere サーバー SQL の使用法』


## -pt dbsrv12 サーバーオプション

パケットの圧縮が適用される最小パケットサイズを増減します。

### 構文

**dbsrv12 -pt** *size* ...

### デフォルト

120 バイト

### 適用対象

すべてのオペレーティングシステムとネットワークサーバー

### 備考

このパラメーターには、圧縮が適用されるパケットサイズの最小バイト数を表す整数値を指定します。80 未満の値はおすすめしません。

状況によっては、圧縮のスレッショルドを変更すると、パケットの転送速度が上昇する場合のみパケットを圧縮できるようになり、圧縮された接続のパフォーマンスが向上することがあります。ほとんどの場合、デフォルト設定が適しています。

クライアントとサーバーで圧縮スレッショルドの設定が異なる場合は、クライアントの設定が適用されます。

### 参照


- 「-p dbeng12/dbsrv12 サーバーオプション」262 ページ
- 「-pc dbsrv12 サーバーオプション」263 ページ
- 「通信の圧縮設定」106 ページ
- 「CompressionThreshold (COMPTH) 接続パラメーター」330 ページ
- 「圧縮の慎重な使用」『SQL Anywhere サーバー SQL の使用法』


## -qi dbeng12/dbsrv12 サーバーオプション

データベースサーバーシステムトレイアイコンとデータベースサーバーメッセージウインドウを表示するかどうかを制御します。

### 適用対象

Windows

### 備考

このオプションは、起動エラーウィンドウを除いて、サーバーの実行中に視覚的な表示が出ないようにします。-o または -oe で指定したログファイルのいずれか (または両方) を使用してエラーを診断できます。

### 参照


- 「-qn dbeng12/dbsrv12 サーバーオプション」265 ページ
- 「-qp dbeng12/dbsrv12 サーバーオプション」266 ページ
- 「-qs dbeng12/dbsrv12 サーバーオプション」266 ページ
- 「-qw dbeng12/dbsrv12 サーバーオプション」267 ページ
- 「-o dbeng12/dbsrv12 サーバーオプション」258 ページ
- 「-oe dbeng12/dbsrv12 サーバーオプション」259 ページ


## -qn dbeng12/dbsrv12 サーバーオプション

起動時にデータベースサーバーメッセージウィンドウを最小化しないことを指定します。

### 構文

{ **dbeng12** | **dbsrv12** } **-qn** ...

### 適用対象

Windows

Linux (X-Window Server が使用されている場合)

### 備考

デフォルトでは、データベースサーバーが起動すると、データベースサーバーメッセージウィンドウは自動的に最小化されます。このオプションを指定すると、データベースサーバーが起動しても、データベースサーバーメッセージウィンドウは最小化されません。

アプリケーションがアクティブでなく、-qn が指定されているときにアプリケーションがデータベースサーバーを自動的に起動する場合、データベースサーバーメッセージウィンドウはバックグラウンドで表示されることがあります。

Linux では、-qn オプションとともに -ux オプション (X-Window Server の使用を指定) を指定する必要があります。

### 参照


- 「-ux dbeng12/dbsrv12 サーバーオプション」286 ページ
- 「-qi dbeng12/dbsrv12 サーバーオプション」264 ページ
- 「-qp dbeng12/dbsrv12 サーバーオプション」266 ページ
- 「-qs dbeng12/dbsrv12 サーバーオプション」266 ページ
- 「-qw dbeng12/dbsrv12 サーバーオプション」267 ページ

### 例

次のコマンドは、Linux または Solaris 上でデータベースサーバーを起動し、データベースサーバーメッセージウィンドウを表示して、データベースサーバーの起動が完了した後でデータベースサーバーメッセージウィンドウを最小化しません。

```
dbeng12 -ux -qn sample.db
```

## -qp dbeng12/dbsrv12 サーバーオプション

データベースサーバーメッセージウィンドウにパフォーマンスに関するメッセージを表示しないことを指定します。

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

データベースサーバーメッセージウィンドウにパフォーマンスに関するメッセージを表示しないようにします。表示されなくなるメッセージには次のものがあります。

●テーブル "*table-name*" にユニークなインデックスまたはプライマリキーがありません。

●データベースファイル "*mydatabase.db*" は *nnn* のディスクフラグメントで構成されています。

### 参照


- 「-qi dbeng12/dbsrv12 サーバーオプション」264 ページ
- 「-qn dbeng12/dbsrv12 サーバーオプション」265 ページ
- 「-qs dbeng12/dbsrv12 サーバーオプション」266 ページ
- 「-qw dbeng12/dbsrv12 サーバーオプション」267 ページ


## -qs dbeng12/dbsrv12 サーバーオプション

起動エラーウィンドウを表示しないようにします。

### 適用対象

Windows

### 備考

このオプションを指定すると、起動エラーウィンドウが表示されなくなります。起動エラーには、たとえばデータベースサーバーでデータベースファイルを開いたり、読み取ったりできない場合や、指定された名前の別のデータベースサーバーがすでに実行中なのでデータベースサーバーが起動しない場合などがあります。

Windows プラットフォームでは、サーバーが自動的に起動しない場合、これらのエラーがウィンドウに表示されるので、これらをクリアしてからサーバーを停止する必要があります。これらのウィンドウは -qs オプションを使用すると表示されません。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

言語 DLL のロード時のエラーの場合、-qs がコマンドラインで指定されていても @data 構文に指定されていないとウィンドウは表示されません。このエラーは -o または -oe で指定されたログには記録されず、Windows アプリケーションイベントログに記録されます (Windows Mobile の場合を除く)。

-qs がコマンドラインで指定されていても、@data 拡張で指定されていない場合、使用法エラーは表示されません。

### 参照


- 「-qi dbeng12/dbsrv12 サーバーオプション」264 ページ
- 「-qn dbeng12/dbsrv12 サーバーオプション」265 ページ
- 「-qp dbeng12/dbsrv12 サーバーオプション」266 ページ
- 「-qw dbeng12/dbsrv12 サーバーオプション」267 ページ
- 「-o dbeng12/dbsrv12 サーバーオプション」258 ページ
- 「-oe dbeng12/dbsrv12 サーバーオプション」259 ページ


## -qw dbeng12/dbsrv12 サーバーオプション

データベースサーバーメッセージウィンドウを表示しないことを指定します。

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

このオプションを指定すると、データベースサーバーメッセージウィンドウが表示されなくなります。Windows プラットフォームでは、データベースサーバーシステムトレイアイコンは表示されます。-o または -oe で指定したログファイルのいずれか (または両方) を使用してエラーを診断できます。

### 参照


- 「-qi dbeng12/dbsrv12 サーバーオプション」264 ページ
- 「-qn dbeng12/dbsrv12 サーバーオプション」265 ページ
- 「-qp dbeng12/dbsrv12 サーバーオプション」266 ページ
- 「-qs dbeng12/dbsrv12 サーバーオプション」266 ページ


## -r dbeng12/dbsrv12 サーバーオプション

データベースサーバー上で起動されるすべてのデータベースを、強制的に読み込み専用にします。データベースへの変更はできません。つまり、データベースサーバーはデータベースファイルやトランザクションログファイルを変更しません。

### 構文

**{ dbeng12 | dbsrv12 } -r** ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

コマンドラインでどのデータベース名よりも前にオプションを指定した場合、テンポラリファイルを除くすべてのデータベースファイルが読み込み専用モードで開きます。あるデータベース名の後ろに -r オプションを指定した場合、そのデータベースだけが読み込み専用になります。テンポラリテーブルに変更を加えることはできますが、トランザクションログとロールバックログが無効化されているため、ROLLBACK を実行しても効果はありません。

変更が不可能なデータベースファイルの例として、CD-ROM によって配布されるデータベースがあります。このようなデータベースには読み込み専用モードでアクセスできます。

INSERT 文や DELETE 文などを使ってデータベースを変更しようとすると、SQLSTATE_READ_ONLY_DATABASE エラーが発生します。

リカバリを必要とするデータベースは、読み込み専用モードでは起動できません。たとえば、オンラインバックアップを使って作成したデータベースファイルは、バックアップの開始時に開いているトランザクションがあると、読み込み専用モードで起動できません。これは、バックアップコピーの開始時に、これらのトランザクションがリカバリを必要とするためです。

監査がオンであるデータベースを読み込み専用モードで起動することはできません。

バックアップコピーの妥当性を検証する場合は、どんな方法でも変更できないように、読み込み専用モードでデータベースを実行してください。「データベースの検証」1118 ページを参照してください。

### 参照


- 「-r dbeng12/dbsrv12 データベースオプション」313 ページ
- 「auditing オプション」598 ページ
- 「読み込み専用メディアでのデータベースの配備」『SQL Anywhere サーバー プログラミング』
- 「特殊モード」160 ページ


### 例

次のコマンドは、2 つのデータベースを読み込み専用モードで起動します。

```
dbeng12 -r database1.db database2.db
```

次のコマンドは、2 つのうち最初のデータベースだけを読み込み専用モードで開きます。

```
dbeng12 database1.db -r database2.db
```

## -s dbeng12/dbsrv12 サーバーオプション

Syslog メッセージのユーザー ID を設定します。

### 構文

{ **dbeng12** | **dbsrv12** } **-s** { **none** | **user** | **daemon** | **local***n* } ...

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 適用対象

UNIX、Mac OS X

### 備考

Syslog 機能へのメッセージに使用されるシステムユーザー ID を設定します。フォアグラウンドで起動しているデータベースサーバーのデフォルトは user で、バックグラウンドで起動しているサーバー (dbspawn で起動した場合、クライアントが自動的に起動した場合、-ud データベースサーバーオプションで起動した場合など) のデフォルトは daemon です。

none を指定すると、Syslog メッセージはログに記録されません。引数 local*n* を使用すると、機能識別子を使用してメッセージをファイルへリダイレクトできます。*n* に 0 ～ 7 の数字を指定できます。詳細については、UNIX Syslog(3) の man ページを参照してください。

**♦ 機能識別子を使用してメッセージをファイルへリダイレクトするには、次の手順に従います。**

次の手順は Solaris でのメッセージのリダイレクト方法を説明したものですが、Linux、IBM AIX、Mac OS X でも使用できます。HP-UX などのプラットフォームでは、*syslog.conf* ファイルは別の場所にあります。*/var/adm/sqlanywhere* ファイルは、任意の場所に配置できます。

1. システムで実行している他のアプリケーションが使用していないユニークな機能識別子を選択します。

   */etc/syslog.conf* ファイルを見ると、local*n* 機能が参照する識別子を確認できます。

2. */etc/syslog.conf* ファイルを開き、次の行を追加します。local*n* は手順 1 で選択した機能識別子です。

   ```
   localn.err;localn.info;localn.notice  /var/adm/sqlanywhere
   ```

3. */var/adm/sqlanywhere* ファイルを作成します。

   ```
   touch /var/adm/sqlanywhere
   ```

4. syslogd のプロセス ID を検索して、*syslog.conf* ファイルを変更したことを syslogd プロセスに通知します。

   ```
   ps -ef | grip syslogd
   ```

   次に、以下のコマンドを実行します。*pid* は syslogd のプロセス ID です。

   ```
   kill -HEAP pid
   ```

5. 次のコマンドを使用してデータベースサーバーを起動します。**local*n*** は手順 1 で選択した機能識別子です。

   ```
   dbeng12 -s localn ...
   ```

   これで、データベースサーバーが Syslog にレポートするメッセージは */var/adm/sqlanywhere* ファイルにリダイレクトされます。

### 参照

- 「MESSAGE 文」『SQL Anywhere サーバー SQL リファレンス』

# -sb dbeng12/dbsrv12 サーバーオプション

ブロードキャストに対するデータベースサーバーの動作を指定します。

### 構文

**{ dbeng12 | dbsrv12 } -sb { 0 | 1 } ...**

### 適用対象

TCP/IP

「クラウドに関する注意点」このオプションは、クラウドのサーバーでは完全にテストされていません。

### 備考

デフォルトで、データベースサーバーは、クライアントアプリケーションから UDP ブロードキャストを受け取り、これらのブロードキャストに適切に応答できるように、1 つ以上の UDP リスナーを起動します。

Embedded SQL 接続に対して、dblocate ユーティリティと db_locate_servers 関数によって、ブロードキャストパケットはすべてのデータベースサーバーを検索する際にローカルネットワーク上に送出されます。データベースサーバー内の UDP リスナーは、その後、データベースサーバーへの接続方法について、情報付きで送信側に応答します。

UDP リスナーは、また、クライアント接続要求に含めることもできます。

-sb オプションは、データベースサーバー内の UDP リスナーの動作を制御します。

-sb 0 を使用すると、データベースサーバーで UDP ブロードキャストリスナーの起動が停止されます。この動作によって、クライアントがデータベースサーバーに接続するときに、Host 接続パラメーターまたは Host プロトコルオプションの使用が強制されます。さらに、サーバーがデフォルトポート (2638) 以外のポートを使用している場合、クライアントもサーバーのポートを指定する必要があります。-sb 0 が指定されると、データベースーサーバーは dblocate からのブロードキャスト要求に応答できなくなり、そのため、データベースサーバーは dblocate 出力には含まれません。

-sb 1 を使用すると、データベースサーバーが dblocate からのブロードキャストに応答しなくなるので、見つからなかったり、リストされなくなったりします。UDP リスナーは、データベースサーバーに接続しようとしているクライアントからの要求に応答できるようにするために、引き続き起動します。

### 参照


- 「BroadcastListener (BLISTENER) プロトコルオプション (サーバー側のみ)」373 ページ
- 「サーバー列挙ユーティリティ (dblocate)」999 ページ
- 「Host 接続パラメーター」346 ページ
- 「Host (IP) プロトコルオプション (クライアント側のみ)」381 ページ


 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## -sf dbeng12/dbsrv12 サーバーオプション

現在のデータベースサーバーで実行されるデータベースの機能を有効または無効にします。

### 構文

{ **dbeng12** | **dbsrv12** } **-sf** *feature-list* ...

### 指定可能な値

次の *feature-name* 値がサポートされます。

- **none**　いずれの機能も無効にしないことを指定します。
- **all**　無効にすることが可能なすべての機能を無効にします。次のグループが含まれます。
  - **client**　クライアント関連の入出力へのアクセスを可能にするすべての機能を無効にします。これには、クライアントコンピューティング環境へのアクセスも含まれます。この機能セットには、次の機能が含まれています。
    - **read_client_file**　クライアントファイルの読み込みを可能にする文の使用を無効にします。たとえば、READ_CLIENT_FILE 関数や LOAD TABLE 文がこれに該当します。「クライアントコンピューター上のデータへのアクセス」『SQL Anywhere サーバー SQL の使用法』を参照してください。
    - **write_client_file**　クライアントファイルへの書き込みを可能にする文の使用を無効にします。たとえば、UNLOAD 文や WRITE_CLIENT_FILE 関数がこれに該当します。「クライアントコンピューター上のデータへのアクセス」『SQL Anywhere サーバー SQL の使用法』を参照してください。
  - **local**　すべての local 機能を無効にします。これには、サーバーコンピューティング環境へのアクセスも含まれます。この機能セットには、次に示す **local_call**, **local_db**、**local_io**、**local_log** 機能サブセットが含まれています。
    - **local_call**　サーバーが所有および制御していないコードの実行を可能にするすべての機能を無効にします。この機能セットには、次の機能が含まれています。
      - **cmdshell**　xp_cmdshell プロシージャーを使用不可にします。「xp_cmdshell システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。
      - **external_procedure**　外部ストアドプロシージャーを使用不可にします。この設定によってデータベースサーバーに組み込まれている xp_* システムプロシージャー (xp_cmdshell や xp_readfile など) が使用できなくなることはありません。これらのシステムプロシージャーには、個別の機能制御オプションがあります。「SQL Anywhere 外部呼び出しインターフェイス」『SQL Anywhere サーバー プログラミング』を参照してください。
      - **java**　Java 関連の機能 (Java プロシージャーなど) を無効にします。「SQL Anywhere で使用する Java クラスの作成」『SQL Anywhere サーバー プログラミング』を参照してください。

- **local_db**　データベースファイル関連のすべての機能を無効にします。この機能セットには、次の機能が含まれています。

  - **backup**　BACKUP 文を使用不可にします。これにより、サーバー側のバックアップを実行できなくなります。クライアント側のバックアップは dbbackup を使用して実行できます。「BACKUP 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

  - **restore**　RESTORE DATABASE 文を使用不可にします。「RESTORE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

  - **database**　CREATE DATABASE 文、ALTER DATABASE 文、DROP DATABASE 文、CREATE ENCRYPTED FILE 文、CREATE DECRYPTED FILE 文、CREATE ENCRYPTED DATABASE 文、CREATE DECRYPTED DATABASE 文を使用不可にします。

  - **dbspace**　CREATE DBSPACE 文、ALTER DBSPACE 文、DROP DBSPACE 文を使用不可にします。

- **local_io**　ファイルとその内容への直接アクセスを可能にするすべての機能を無効にします。この機能セットには、次の機能が含まれています。

  - **read_file**　ローカルファイルの読み込みを可能にする文の使用を無効にします。たとえば、xp_read_file システムプロシージャー、LOAD TABLE 文、OPENSTRING( FILE ...) の使用がこれに該当します。). 代替名の load_table や xp_read_file は廃止されました。

  - **write_file**　ローカルファイルへの書き込みを可能にする文の使用を無効にします。たとえば、UNLOAD 文や xp_write_file システムプロシージャーがこれに該当します。代替名の unload_table や xp_write_file は廃止されました。

  - **delete_file**　ローカルファイルの削除を可能にする文の使用を無効にします。たとえば、データベースファイルを削除する db_delete_file DBLib 関数の使用を無効にします。db_delete_file 関数は dbbackup -x オプションと -xo オプションで使用されるため、db_delete_file を無効にすると、-x オプションや -xo オプションが指定されている場合、dbbackup の実行は失敗します。「db_delete_file 関数」『SQL Anywhere サーバー プログラミング』を参照してください。

  - **directory**　ディレクトリクラスプロキシテーブルを使用不可にします。この機能は、remote_data_access をオフに設定した場合にも無効になります。

- **local_log**　ディスク上にファイルを作成し、データを直接書き込む、すべてのロギング機能を無効にします。この機能セットには、次の機能が含まれています。

  - **request_log**　要求ログのファイル名を変更する機能と、その最大サイズまたは最大ファイル数を増加する機能を無効にします。データベースサーバーの起動コマンドには、要求ログファイルとそのファイルの最大サイズを指定できます。ただし、それらをサーバーの起動後に変更することはできません。要求ログ機能を無効にしても、要求ロギングのオンとオフを切り替えたり、要求ログファイルの最大サ


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

イズや最大数を小さくしたりすることは可能です。「要求ロギング」『SQL Anywhere サーバー SQL の使用法』を参照してください。

○ **console_log** sa_server_option システムプロシージャーの ConsoleLogFile オプションを使用してデータベースサーバーメッセージログのファイル名を変更する機能を無効にします。また、sa_server_option システムプロシージャーの ConsoleLogMaxSize オプションを使用してログファイルの最大サイズを変更する機能を無効にします。データベースサーバーの起動時には、サーバーログファイルとそのサイズを指定できます。

○ **webclient_log** sa_server_option システムプロシージャーの WebClientLogFile オプションを使用して Web サービスクライアントログのファイル名を変更する機能を無効にします。データベースサーバーの起動時に Web サービスクライアントログファイルを指定できます。「-zoc dbeng12/dbsrv12 サーバーオプション」300 ページを参照してください。

○ **remote** リモートアクセスまたはリモートプロセスとの通信を可能にするすべての機能を無効にします。この機能セットには、次の機能が含まれています。

● **remote_data_access** プロキシテーブルなどのリモートデータアクセスサービスを使用不可にします。

● **send_udp** sa_send_udp システムプロシージャーを使用して指定したアドレスに UDP パケットを送信する機能を無効にします。

● **web_service_client** Web サービスクライアントのストアドプロシージャーコール (HTTP 要求を発行するストアドプロシージャー) の使用を無効にします。

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

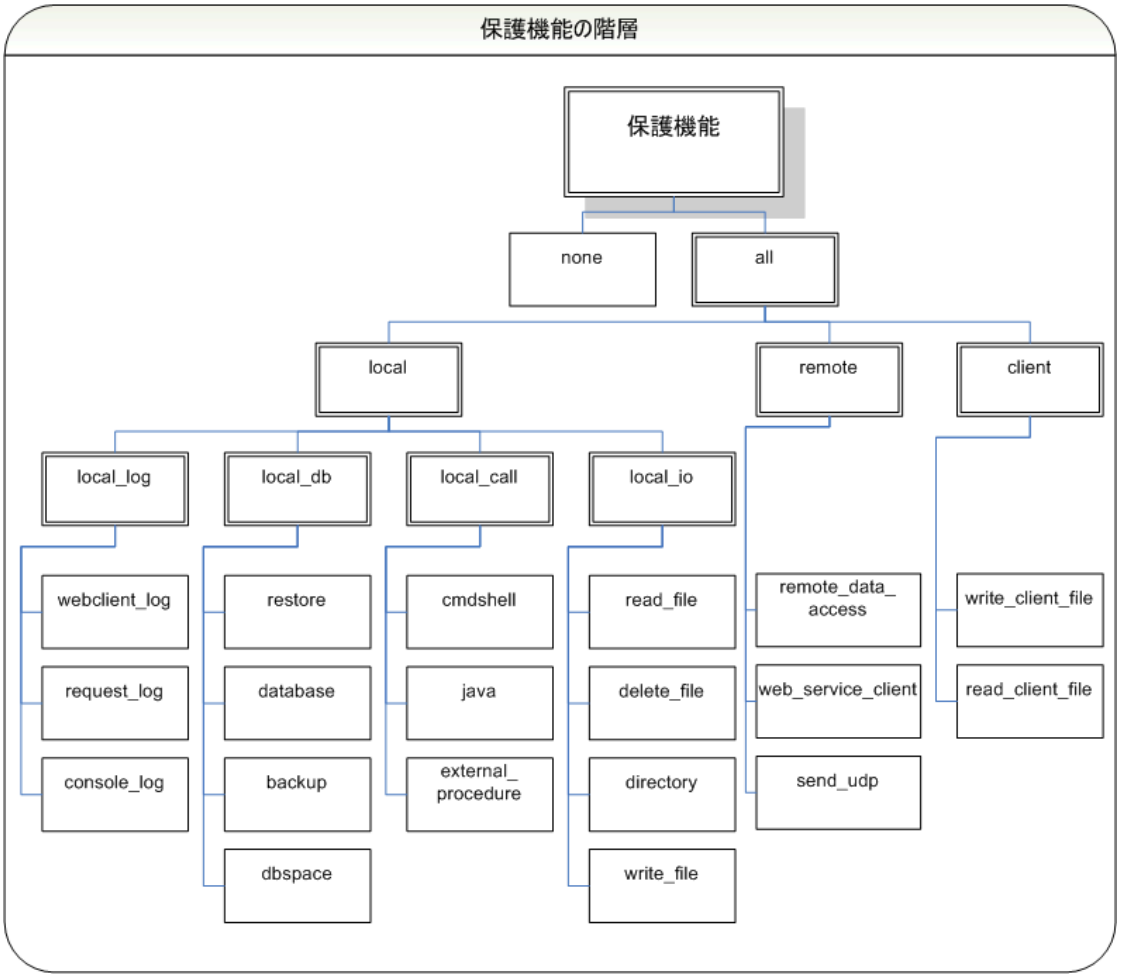
このオプションでは、データベースサーバーの機能を有効または無効にできます。この設定は、データベースサーバー上のすべてのデータベースに影響します。-sk オプションで指定されたキーを secure_feature_key オプションに設定することによって、無効になっている (保護されている) すべての接続機能を有効化できます。secure_feature_key オプションが -sk によって指定されたキーに設定された接続では、sa_server_option システムプロシージャーの **SecureFeatures** プロパティを使用して、データベースサーバーで保護されている機能セットを変更することもできます。

*feature-list* は、データベースサーバーで保護する機能名または機能セットをカンマで区切って示したリストです。*feature-name* は機能を無効にすることを示し、*-feature-name* は無効化された機能のリストからその機能を削除することを示します。たとえば、次のコマンドは、DB 領域機能だけを有効にします。

```
dbeng12 -n secure_server -sf all,-dbspace
```

### 機能セット階層

次の図は、すべての機能セットのキーワードと階層構造を示しています。たとえば、**local_io** は **read_file**、**write_file**、**delete_file**、**directory** 機能を包含しています。




### 参照

- 「-sk dbeng12/dbsrv12 サーバーオプション」275 ページ
- 「secure_feature_key」677 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「機能の保護」1387 ページ

### 例

次のコマンドは、要求ログとすべてのリモートデータアクセス機能へのアクセスを無効にしてデータベースサーバー secure_server を起動します。これらの機能は、-sk で指定したキーを secure_feature_key データベースオプションに設定して、特定の接続において有効にできます。

```
dbsrv12 -n secure_server -sf request_log,remote -sk j978kls12
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

secure_server データベースサーバー上のデータベースに接続するときに secure_feature_key オプションを -sk で指定された値に設定すると、その接続において、要求ログへのアクセスとリモートデータアクセス機能が有効になります。

```
SET TEMPORARY OPTION secure_feature_key = 'j978kls12';
```

次のコマンドは、ローカルデータベース機能を除き、すべての機能を無効にします。

```
dbeng12 -n secure_server -sf all,-local_db
```

## -sk dbeng12/dbsrv12 サーバーオプション

データベースサーバーで無効になっている機能を有効にするためのキーを指定します。

### 構文

{ **dbeng12** | **dbsrv12** } **-sk** *key* ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

-sf オプションを使用してデータベースサーバーの機能を保護するときには、-sk オプションを使用して、機能を有効にするキーも指定できます。このキーを secure_feature_key データベースオプションに指定すると、保護されている機能が有効になります。この場合、sa_server_option システムプロシージャーを使用して、データベースサーバー上で実行されているすべてのデータベースを保護する機能または機能セットに変更を加えることもできます。

secure_feature_key オプションを -sk で指定された値以外の値に設定すると、エラーは発生せず、-sf で指定された機能はその接続において保護されたままとなります。

### 参照


- 「-sf dbeng12/dbsrv12 サーバーオプション」271 ページ
- 「secure_feature_key」677 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「機能の保護」1387 ページ


### 例

次のコマンドは、バックアップ機能へのアクセスを無効にしてデータベースサーバー secure_server を起動します。これらの機能は、-sk で指定したキーを後で使用して、特定の接続において有効にできます。

```
dbsrv12 -n secure_server -sf backup -sk j978kls12
```

secure_server データベースサーバー上のデータベースに接続するときに secure_feature_key オプションを -sk で指定された値に設定すると、バックアップの実行が可能となり、secure_server データベースサーバー上で無効になっている機能を変更できます。

```
SET TEMPORARY OPTION secure_feature_key = 'j978kls12';
```

その後、次のコマンドを実行して、secure_server データベースサーバー上で実行されているデータベースに対して、保護されたすべての機能を無効にできます。

```
CALL sa_server_option( 'SecureFeatures', 'all' );
```

## -su dbeng12/dbsrv12 サーバーオプション

ユーティリティデータベース (utility_db) の DBA ユーザーのパスワードを設定します。または、ユーティリティデータベースへの接続を無効にします。

### 構文

{ **dbeng12** | **dbsrv12** } **-su** *password* ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

このオプションは、ユーティリティデータベースの DBA ユーザーの最初のパスワードを指定します。パスワードでは大文字と小文字が区別されます。ユーティリティデータベースへのすべての接続を無効にするには、パスワードに **none** を指定します。ユーティリティデータベースのパスワードをコマンドライン上でクリアテキスト入力を回避するには、dbfhide を使用してパスワードを含むファイルの内容を読みにくくし、難読化されたファイルをコマンドライン上で参照します。

- パーソナルデータベースサーバーを使用している場合、-su オプションを指定しないと、ユーザー ID = DBA と任意のパスワードを使用してユーティリティデータベースに接続できます。

- ネットワークデータベースサーバーを使用している場合で、-su オプションを指定しないときに、ユーティリティデータベースに接続するには、*util_db.ini* ファイルが存在することと、ユーザー ID = DBA および *util_db.ini* ファイルの内容と一致するパスワードを使用することが必要です。ネットワークサーバーで -su と *util_db.ini* ファイルの両方を使用すると、*util_db.ini* ファイルが無視されます。*util_db.ini* ファイルの使用は推奨されません。

  ネットワークデータベースサーバー用のユーティリティデータベースにパスワードを指定することは、データベースサーバーが実行中の場合には便利ですが、データベースに接続できません。たとえば、ミラーリングシステムでは、ユーティリティデータベースに接続してデータベースサーバーをシャットダウンしたり、必要に応じてミラーサーバーを強制的にプライマリサーバーにしたりできるようになります。「ミラーリングシステムのデータベースサーバーの停止」1168 ページと「データベースサーバーの強制プライマリサーバー化」1166 ページを参照してください。

utility_db に接続しているときに、CREATE USER DBA IDENTIFIED BY *new-password* 文を実行して、ユーティリティデータベースの DBA ユーザーのパスワードを変更できます。utility_db データベースへの接続を無効にするには、REVOKE CONNECT FROM DBA 文を使用します。「REVOKE 文」『SQL Anywhere サーバー SQL リファレンス』と「ユーティリティデータベースに使用できる文」136 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照


- 「ユーティリティデータベース」135 ページ
- 「ユーティリティデータベース接続」137 ページ
- 「ファイル非表示ユーティリティ (dbfhide)」953 ページ
- 「CREATE USER 文」『SQL Anywhere サーバー SQL リファレンス』
- 「REVOKE 文」『SQL Anywhere サーバー SQL リファレンス』


### 例

次のコマンドは、ユーティリティデータベースへのすべての接続を無効にします。

```
dbeng12 -su none c:¥inventory.db
```

次の例では、ユーティリティデータベースのパスワードを含む *util_db_pwd.cfg* ファイルが dbfhide によって難読化され、ファイル名が *util_db_pwd_hide.cfg* に変更されます。

```
dbfhide util_db_pwd.cfg util_db_pwd_hide.cfg
```

その後、*util_db_pwd_hide.cfg* ファイルを使用して、ユーティリティデータベースのパスワードを指定できます。

```
dbsrv12 -su @util_db_pwd_hide.cfg -n my_server c:¥inventory.db
```

## -ti dbeng12/dbsrv12 サーバーオプション

非アクティブ接続を切断します。

### 構文

{ **dbeng12** | **dbsrv12** } **-ti** *minutes* ...

### デフォルト

240 (4 時間)

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

*minutes* で指定された分数の間、要求を送信しなかった接続を切断します。最大値は 32767 です。データベーストランザクション処理中のクライアントコンピューターは、トランザクションが終了するか、接続が切断されるまでロックされます。-ti オプションを指定すると、非アクティブ接続が切断され、ロックが解除されます。

ほとんどの接続はネットワークリンク (TCP) 経由で行われているため、-ti オプションは dbsrv12 と一緒に使用すると非常に便利です。

-ti オプションは、ローカル TCP/IP 接続の場合のみ dbeng12 と 一緒に使用すると便利です。-ti を使用しても、共有メモリを使用しているローカルサーバーへの接続には影響しませんが、共有メモリ接続に適用される接続にタイムアウトを指定できます。デフォルトでは、パーソナルデータ

ベースサーバーは TCP/IP を起動しません。「-x dbeng12/dbsrv12 サーバーオプション」289 ページを参照してください。

値を 0 に設定すると、非アクティブ接続は検査されず、接続が切断されません。Idle 接続パラメーターが使用されていない場合、TCP/IP 接続のアイドルタイムアウト値は -ti データベースサーバーオプションによって制御されます。両方の -ti データベースサーバーオプションと Idle 接続パラメーターが指定された場合、アイドルタイムアウト値は接続パラメーターによって制御されます。

### 参照


- IdleTimeout サーバープロパティ761 ページ
- 「-tl dbeng12/dbsrv12 サーバーオプション」278 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「タイムアウト値の調整」1215 ページ
- 「Idle 接続パラメーター」348 ページ


## -tl dbeng12/dbsrv12 サーバーオプション

活性パケットを送信する期間を設定します。

### 構文

**{ dbeng12 | dbsrv12 } -tl** *seconds* ...

### 適用対象

TCP/IP を使用するすべてのデータベースサーバー

「クラウドに関する注意点」このオプションは、クラウドのサーバーでは完全にテストされていません。

### 備考

接続が維持されていることを確認するため、クライアント／サーバーの TCP/IP 通信プロトコルを介して、定期的に活性パケットが送信されます。接続で活性パケットを検出することなく、指定した LivenessTimeout 時間 (デフォルトは 2 分) にわたってサーバーが実行されていると、通信は切断され、サーバーはそのクライアントに関連付けられている接続を削除します。非スレッドの UNIX クライアントと TDS 接続では、活性パケットによる確認は行われません。

サーバーで -tl オプションを指定すると、活性期間が指定されていないすべてのクライアントに対して LivenessTimeout 値を設定できます。

LivenessTimeout 値の 3 分の 1 から 3 分の 2 の期間で接続がパケットを送信しない場合に、活性パケットが送信されます。

接続数が 200 を超えると、サーバーは指定されている LivenessTimeout 値に基づいて、それより大きい LivenessTimeout 値を自動的に計算するため、多数の接続を効率的に処理できます。活性パケットは、各アイドル接続において、LivenessTimeout 値の 3 分の 1 から 3 分の 2 の期間で送信されます。大量の活性パケットが同時に送信されることはありません。送信に時間がかかる

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

活性パケットは、LivenessTimeout 値の 3 分の 2 の期間が経過した後で送信できます。活性パケットの送信に時間がかかる場合、データベースサーバーメッセージログに警告が表示されます。この警告が発生したら、LivenessTimeout 値の増加を検討してください。

これは一般的にはおすすめしませんが、次のように指定して活性タイムアウトを無効にできます。

```
dbsrv12 -tl 0 -n my_server
```

LivenessTimeout オプションを無効にせずに、次のように値を 1 時間に増やすことを検討してください。

```
dbsrv12 -tl 3600 -n my_server
```

### 参照


- LivenessTimeout サーバープロパティ 763 ページ
- 「-ti dbeng12/dbsrv12 サーバーオプション」 277 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「タイムアウト値の調整」 1215 ページ


## -tmf dbeng12/dbsrv12 サーバーオプション

トランザクションマネージャーに、強制的に分散トランザクションをリカバリさせます。

### 構文

**{ dbeng12 | dbsrv12 } -tmf** ...

### 適用対象

Windows

### 備考

このオプションは、分散トランザクションコーディネーターが使用できない場合に、分散トランザクションのリカバリ中に使用します。DTC が特定できないときは、未処理の操作をロールバックしてリカバリを続行します。また、DTC が使用できないプラットフォームで、トランザクションログに分散トランザクションがあるデータベースを起動する場合にも使用できます。

> **警告**
> このオプションを使用すると、分散トランザクションは正常にはリカバリされません。日常的な使用は想定していません。

### 参照


- 「-tmt dbeng12/dbsrv12 サーバーオプション」 280 ページ
- 「分散トランザクションからのリカバリ」『SQL Anywhere サーバー プログラミング』

## -tmt dbeng12/dbsrv12 サーバーオプション

分散トランザクションに参加するための再エンリストタイムアウトを設定します。

### 構文

{ **dbeng12** | **dbsrv12** } **-tmt** *milliseconds* ...

### 適用対象

Windows

### 備考

分散トランザクションのリカバリ時に使用します。この値は、データベースサーバーが再登録されるまでの待機時間を指定します。デフォルトでは、タイムアウトはありません (データベースサーバーは、無期限に待機します)。

### 参照


- 「-tmf dbeng12/dbsrv12 サーバーオプション」279 ページ
- 「分散トランザクションからのリカバリ」『SQL Anywhere サーバー プログラミング』


## -tq dbeng12/dbsrv12 サーバーオプション

指定の時刻にサーバーを停止します。

### 構文

{ **dbeng12** | **dbsrv12** } **-tq** { *datetime* | *time* } ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

このオプションは、自動オフラインバックアッププロシージャーの設定に役立ちます。「バックアップとデータリカバリ」1071 ページを参照してください。

時間のフォーマットは *hh*:*mm* (24 時間表記) で、オプションで前に日付を付けることができます。日付を指定する場合は、日付と時間を二重引用符で囲んで、"*YYYY/MM/DD HH*:*MM*" の形式にする必要があります。

### 参照


- QuittingTime サーバープロパティ773 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』

## -u dbeng12/dbsrv12 サーバーオプション

オペレーティングシステムのディスクキャッシュを使用してファイルを開きます。

### 構文

{ **dbeng12** | **dbsrv12** } **-u** ...

### 適用対象

Windows、UNIX

「クラウドに関する注意点」このオプションは、クラウドのサーバーでは完全にテストされていません。

### 備考

データベースキャッシュに加え、オペレーティングシステムのディスクキャッシュも使ってファイルが開かれます。

オペレーティングシステムのディスクキャッシュによってパフォーマンスが向上することもありますが、データベースキャッシュだけを使用すると、パフォーマンスがさらに向上します。

サーバーを専用コンピューターで実行している場合は、-u オプションを使用しないでください。通常、データベースキャッシュを使用する方が効率的です。-u オプションを使用するのは、サーバーを実行するコンピューターに他にもいくつかのアプリケーションがインストールされていて (サイズの大きなデータベースキャッシュが他のアプリケーションを実行するための障害となり)、しかも I/O の多いタスクをサーバー上で断続的に実行する (キャッシュサイズを大きくすることでパフォーマンスが向上する) 場合です。

## -ua dbeng12/dbsrv12 サーバーオプション

非同期 I/O の使用をオフにします。

### 構文

{ **dbeng12** | **dbsrv12** } **-ua** ...

### 適用対象

Linux

### 備考

デフォルトで、使用可能な場合にデータベースサーバーは Linux で非同期 I/O を使用します。非同期 I/O を使用するには、以下の条件を満たす必要があります。

1. ライブラリ *libaio.so* が実行時にロードできる。
2. カーネルが非同期 I/O をサポートしている。

非同期 I/O の使用をオフにしたい場合、-ua dbeng12/dbsrv12 サーバーオプションを指定します。

## -uc dbeng12/dbsrv12 サーバーオプション

データベースサーバーをシェルモードで起動します。これはデフォルトです。

### 構文

**{ dbeng12 | dbsrv12 } -uc** ...

### 適用対象

UNIX、Mac OS X

### 備考

データベースサーバーをシェルモードで起動します。-uc、-ui、-um、-ux のうち 1 つだけを指定してください。-uc を指定すると、データベースサーバーはソフトウェアの以前のリリースと同じ方法で起動されます。

デーモンとしてのデータベースサーバーの起動の詳細については、「-ud dbeng12/dbsrv12 サーバーオプション」282 ページを参照してください。

### 参照


- 「-ui dbeng12/dbsrv12 サーバーオプション」284 ページ
- 「-um dbeng12/dbsrv12 サーバーオプション」284 ページ
- 「-ux dbeng12/dbsrv12 サーバーオプション」286 ページ


## -ud dbeng12/dbsrv12 サーバーオプション

デーモンとして実行します。

### 構文

**{ dbeng12 | dbsrv12 } -ud** ...

### 適用対象

UNIX、Mac OS X

### 備考

このオプションを使用すると、現在のユーザーセッションが終了しても引き続きサーバーを実行できます。

-ud オプションを使用してデーモンを直接起動したときは、dbeng12 コマンドと dbsrv12 コマンドがデーモンプロセスを作成し、(終了して次のコマンドを実行できるように) すぐに返します。その後、デーモンがそれ自体を初期化するか、コマンドで指定されたデータベースを開こうとします。

-ud オプションの代わりに dbspawn を使用することの利点は、デーモンが起動し、要求を受け入れる状態になったことを確認するまで dbspawn プロセスが終了しないことです。何らかの理由でデーモンの起動が失敗した場合、dbspawn の終了コードは 0 以外の値になります。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

データベースサーバーをデーモンとして起動すると、現在のユーザーの unmask 設定によってパーミッションが制御されます。データベースサーバーに適切なパーミッションを付与するため、データベースサーバーを起動する前に unmask 値を設定することをおすすめします。

このオプションと -um オプションと -ud オプションを同時に使用しないでください。

### 参照


- 「サーバーバックグラウンド起動ユーティリティ (dbspawn)」1022 ページ
- 「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』
- 「現在のセッション外でのデータベースサーバーの起動」170 ページ
- 「セキュリティに関するヒント」1380 ページ


## -uf dbeng12/dbsrv12 サーバーオプション

致命的なエラーの発生時に実行するアクションを指定します。

### 構文

{ **dbeng12** | **dbsrv12** } **-uf** *action* ...

### デフォルト

default

### 指定可能な値

- **abort**　UNIX の abort 関数が呼び出され、コアファイルが生成されます。
- **default**　データベースサーバーは、常に abort と同じ動作をします。ただし、デバイスフルの致命的なエラーが発生した場合は、defunct と同じ動作となります。このアクションによって、システムはフルデバイスとなってもコアファイルを作成しようとしません。これはデフォルトの動作です。
- **defunct**　データベースサーバーは実行を続け、abort を呼び出しません。データベースサーバーに接続しようとすると、元の致命的なエラーの SQL エラーを受け取ります。

### 適用対象

UNIX、Mac OS X

### 参照


- 「-oe dbeng12/dbsrv12 サーバーオプション」259 ページ
- 「サポートユーティリティ (dbsupport)」1027 ページ
- 「SQL Anywhere のエラーレポート」1210 ページ
- 「データベースサーバー動作のロギング」153 ページ

## -ui dbeng12/dbsrv12 サーバーオプション

Linux では、このオプションを使用すると、**[サーバー起動オプション]** ウィンドウが開き、データベースサーバーメッセージウィンドウが表示され、X Window Server が起動するかどうかにかかわらず、データベースサーバーが起動します。Mac OS X では、-ui を使用すると、データベースサーバーメッセージが新しいウィンドウに表示され、使用可能な表示がない場合はデータベースサーバーがシェルモードで起動します。

### 構文

**{ dbeng12 | dbsrv12 } -ui** ...

### 適用対象

X-Window Server がサポートされている Linux、Mac OS X

### 備考

Linux では、-ui オプションを使用すると、**[サーバー起動オプション]** ウィンドウを使用して、データベースサーバー起動時のサーバーオプションを指定し、データベースサーバーが起動したらデータベースサーバーメッセージウィンドウを表示できます。Mac OS X では、サーバーメッセージが *DBLauncher.app* 内の新しいウィンドウにリダイレクトされます。

Linux では、-ui dbeng12/dbsrv12 サーバーオプションのみを指定した場合は、**[サーバー起動オプション]** ウィンドウが表示されるので、データベースサーバーの起動オプションを入力できます。Mac OS X では、-ui オプションと、データベースサーバーの起動に必要なその他のオプションを同時に使用する必要があります。

-ui を指定すると、データベースサーバーは使用可能な表示を探そうとします。DISPLAY 環境変数が設定されていなかったり、X-Window Server が実行されていなかったりしたために、使用可能な表示が見つからなかった場合は、データベースサーバーはシェルモードで起動されます。使用可能な表示が見つからない場合にデータベースサーバーの起動を中止するには、-ui オプションではなく -ux オプションを指定します。-uc、-ui、-um、-ux のうち 1 つだけを指定してください。

デーモンとしてのデータベースサーバーの起動の詳細については、「-ud dbeng12/dbsrv12 サーバーオプション」282 ページを参照してください。

### 参照


- 「-uc dbeng12/dbsrv12 サーバーオプション」282 ページ
- 「-um dbeng12/dbsrv12 サーバーオプション」284 ページ
- 「-ux dbeng12/dbsrv12 サーバーオプション」286 ページ


## -um dbeng12/dbsrv12 サーバーオプション

*DBLauncher.app* 内の新しいウィンドウにデータベースサーバーメッセージを表示します。

### 構文

**{ dbeng12 | dbsrv12 } -um** ...

### 適用対象

Mac OS X

### 備考

-um オプションを使用して、*DBLauncher.app* インスタンスが実行中の場合にこのインスタンスに接続し、*DBLauncher.app* 内の新しいウィンドウにメッセージを表示することができます。-um オプションは、データベースサーバーの起動に必要なその他のオプションと同時に使用する必要があります。サーバーメッセージは、シェルではなくこのウィンドウに表示されます。このウィンドウを閉じるとデータベースサーバーが停止します。*DBLauncher.app* インスタンスへの接続を確立できなかった場合、データベースサーバーは起動しません。

データベースサーバーが *DBLauncher.app* インスタンスに接続するには、両方が同じ Mac OS X セキュリティコンテキストで実行されている必要があります。たとえば、SSH セッションから起動されたデータベースサーバーでは、Launch Services によって起動された *DBLauncher.app* インスタンスは認識できません。

このオプションと -um オプションと -ud オプションを同時に使用しないでください。

デーモンとしてのデータベースサーバーの起動の詳細については、「-ud dbeng12/dbsrv12 サーバーオプション」282 ページを参照してください。

### 参照


- 「-uc dbeng12/dbsrv12 サーバーオプション」282 ページ
- 「-ui dbeng12/dbsrv12 サーバーオプション」284 ページ


## -ut dbeng12/dbsrv12 サーバーオプション

テンポラリファイルをタッチします。

### 構文

{ **dbeng12** | **dbsrv12** } **-ut** *minutes* ...

### デフォルト

デフォルトは 30 分です。

### 適用対象

UNIX、Mac OS X

### 備考

このオプションを使用すると、指定の間隔でサーバーにテンポラリファイルをタッチさせることができます。

# -ux dbeng12/dbsrv12 サーバーオプション

Linux で **[サーバー起動オプション]** ウィンドウを開くか、データベースサーバーメッセージウィンドウを表示します (X-Window Server を使用)。

## 構文

{ **dbeng12** | **dbsrv12** } **-ux** ...

## 適用対象

X-Window Server がサポートされている Linux

## 備考

-ux オプションを使用すると、データベースサーバーの起動時に 2 つの操作が可能です。それは、**[サーバー起動オプション]** ウィンドウを使用してデータベースサーバーの起動時にサーバーオプションを指定することと、サーバーが起動した後にデータベースサーバーメッセージウィンドウを表示することです。

-ux dbeng12/dbsrv12 サーバーオプションのみを指定した場合は、**[サーバー起動オプション]** ウィンドウが表示されるので、データベースサーバーの起動オプションを入力できます。

-ux が指定されている場合、サーバーは使用可能な表示を見つけます。DISPLAY 環境変数が設定されていなかったり、X-Window Server が実行されていなかったりしたために、使用可能な表示が見つからなかった場合は、データベースサーバーを起動できません。使用可能な表示が見つからない場合でもデータベースサーバーを起動するには、-ux の代わりに -ui オプションを使用します。

-ux の他にもサーバーオプションを指定した場合は、データベースサーバーが起動するとデータベースサーバーメッセージウィンドウが表示されます。-uc、-ui、-ux のうち 1 つだけを指定してください。

デーモンとしてのデータベースサーバーの起動の詳細については、「-ud dbeng12/dbsrv12 サーバーオプション」282 ページを参照してください。

## 参照


- 「-uc dbeng12/dbsrv12 サーバーオプション」282 ページ
- 「-ui dbeng12/dbsrv12 サーバーオプション」284 ページ
- 「-qn dbeng12/dbsrv12 サーバーオプション」265 ページ


## 例

次のコマンドを入力すると、データベースサーバーの起動オプションを入力するための **[サーバー起動オプション]** ウィンドウが表示されます。

```
dbeng12 -ux
```

次のコマンドを入力すると、データベースサーバーが起動され、データベースサーバーメッセージウィンドウが表示されます。

```
dbeng12 -ux sample.db
```

## -v dbeng12/dbsrv12 サーバーオプション

ソフトウェアのバージョンを表示します。

### 構文

**{ dbeng12 | dbsrv12 } -v** ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

ウィンドウにデータベースサーバーのバージョンを表示して、停止します。

## -vss dbeng12/dbsrv12 サーバーオプション

ボリュームシャドウコピーサービス (VSS) を有効または無効にします。

### 構文

**{ dbeng12 | dbsrv12 } -vss{ + | - }** ...

### 適用対象

32 ビットの Microsoft Windows XP、Microsoft Windows 2003 以降の 32 ビット版と 64 ビット版のオペレーティングシステム

「クラウドに関する注意点」このオプションは、クラウドのサーバーでは完全にテストされていません。

### 備考

デフォルトでは、SQL Anywhere VSS ライター (*dbvss12.exe*) が実行されている場合、すべての SQL Anywhere データベースでバックアップに VSS サービスを使用できます。VSS は、SQL Anywhere VSS ライターなしでデータベースのバックアップに使用できます。ただし、これらのデータベースをリストアするには、SQL Anywhere の完全なリカバリ手順を使用する必要がある可能性があります。データベースサーバーが VSS サービスに参加しないようにするには、データベースサーバーの起動時に **-vss-** を指定します。

### 参照


- 「SQL Anywhere ボリュームシャドウコピーサービス (VSS)」1084 ページ
- 「Windows 用サービスユーティリティ (dbsvc)」1013 ページ
- 「データのメディア障害からのリカバリ」1098 ページ


### 例

次のコマンドは、*mydatabase.db* データベースを起動し、ライター (*dbvss12.exe*) が実行されていても VSS 処理に参加しないようにデータベースサーバーに指定します。

```
dbsrv12 -vss- mydatabase.db
```

# -wc dbeng12/dbsrv12 サーバーオプション

チェックサムがデフォルトでデータベースによって有効にされていない場合、このデータベースサーバーのすべてのデータベースに対する書き込み操作でチェックサムを有効にするかどうかを制御します。

### 構文

**{ dbeng12 | dbsrv12 } -wc[ + | - ] ...**

### 適用対象

すべてのオペレーティングシステムとサーバー。

「クラウドに関する注意点」このオプションは、クラウドのサーバーでは完全にテストされていません。

### 備考

書き込みチェックサム (-wc オプションで有効) とグローバルチェックサム (チェックサムが有効な状態でデータベースを作成) の違いは、-wc ではデータベースページがディスクに書き込まれるときにのみチェックサムが行われる点です。ディスクから読み込まれるページは、ページが書き込まれる前にチェックサム値が計算された場合にのみ検証されます。データベースでチェックサムが有効になっている場合、ページが書き込まれるときにすべてのページに対してチェックサムが計算され、ページが読み込まれるときにすべてのページに対してチェックサムが検証されます。

データベースが Windows Mobile またはリムーバブルストレージデバイス (ネットワーク共有や USB デバイスなど) で実行されていることがデータベースサーバーで検出されると、データベースサーバーでは、すべてのデータベースページに対してグローバルチェックサムが自動的に有効になります。

デフォルトでは、バージョン 10 または 11 の SQL Anywhere で作成されたデータベースではグローバルチェックサムは有効になっていません。バージョン 12 以降のデータベースサーバーで SQL Anywhere 11 を使用して作成されたデータベースを起動すると、デフォルトでは、ページがディスクに書き込まれるときにデータベースサーバーでページに対して書き込みチェックサムが作成されます (-wc+)。バージョン 12 以降のデータベースではデフォルトでグローバルチェックサムが有効になっています。デフォルトではすべてのデータベースページにチェックサムがあるため、これらのデータベースに対してデータベースサーバーのデフォルト値は -wc- になります。デフォルトのチェックサム設定を使用しない場合は、-wc オプションまたは START DATABASE 文を使用してデータベースサーバーのチェックサム動作を変更できます。

次の文を実行することによって、データベースがグローバルチェックサムを有効にして作成されたかどうかをチェックできます。

```
SELECT DB_PROPERTY ( 'Checksum' );
```

次の文を実行することによってのみ、書き込み I/O 操作に対して書き込みチェックサムが有効になっているかどうかをチェックできます。

```
SELECT DB_PROPERTY ( 'WriteChecksum' );
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照


- 「チェックサムを使用した破損の検出」1120 ページ
- 「-wc dbeng12/dbsrv12 サーバーオプション」288 ページ
- 「START DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』


## -x dbeng12/dbsrv12 サーバーオプション

サーバー側のネットワーク通信プロトコルを指定します。

### 構文

{ **dbeng12** | **dbsrv12** } **-x** { **all** | **none** | *server-protocols* } ...

*server-protocols*:
{ **tcpip** *parmlist* },...
*parmlist*:
**(** *parm*=*value*;...**)**

### 指定可能な値

-x オプションについて選択した設定に関係なく、データベースサーバーは常に共有メモリ接続を受け入れます。

デフォルトでは、パーソナルデータベースサーバーは共有メモリプロトコルのみを起動しますが、ネットワークデータベースサーバーは共有メモリプロトコルと TCP/IP プロトコルを起動します。

また、-x オプションに次の値も指定できます。

- **TCPIP (TCP)**　共有メモリプロトコルと TCP/IP プロトコルを使用するクライアントによる接続試行を受信します。TCP/IP プロトコルは、ネットワークサーバーと、同一コンピューター通信用のパーソナルデータベースサーバーでサポートされています。
- **NONE**　共有メモリプロトコルのみを使用するクライアントによる接続要求を受信します。
- **ALL**　共有メモリプロトコルと TCP/IP プロトコルを使用するクライアントによる接続試行を受信します。

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

-x オプションを使用して、クライアント接続の受信に使用する通信プロトコルを指定します。

UNIX での共有メモリ接続の保護については、「セキュリティに関するヒント」1380 ページを参照してください。

TCP/IP プロトコルの場合、追加のパラメーターは次のフォーマットで提供されます。

```
-x tcpip(PARM1=value1;PARM2=value2;...)
```

使用可能なパラメーターの詳細については、「ネットワークプロトコルオプション」369 ページを参照してください。

UNIX では、複数のパラメーターを指定する場合、または丸カッコなどの特定の記号文字を使用する場合に二重引用符が必要です。次に例を示します。

```
-x "tcpip(PARM1=value1;PARM2=value2;...)"
```

データベースサーバーが TCP/IP 接続を受信するとき、デフォルトでポート 2638 からすべてのネットワークカードを受信します。「ネットワークプロトコルオプション」369 ページを参照してください。

### 参照


- 「データベースサーバーの起動例」145 ページ
- 「-xd dbeng12/dbsrv12 サーバーオプション」292 ページ
- 「-xp dbsrv12 データベースオプション」318 ページ
- 「-xs dbeng12/dbsrv12 サーバーオプション」294 ページ
- 「CommLinks (LINKS) 接続パラメーター」327 ページ
- 「TCP/IP プロトコル」99 ページ


### 例

ネットワークサーバーで次のように共有メモリ接続と TCP/IP 接続を許可します。

```
dbsrv12 -n server_name "%SQLANYSAMP12%¥demo.db"
```

-x オプションを使用して、TCP/IP の動作を調整するプロトコルオプションを指定します。ネットワークサーバーに 2 つの具体的なネットワークカードを使用することを許可します。

```
dbsrv12 -x "tcpip(MyIP=192.75.209.12,192.75.209.32)" "%SQLANYSAMP12%¥demo.db"
```

パーソナルサーバーで次のように共有メモリ接続と TCP/IP 接続を許可します。

```
dbeng12 -x tcpip "%SQLANYSAMP12%¥demo.db"
```

## -xa dbsrv12 サーバーオプション

監視サーバーに対するデータベース名と認証文字列のカンマ区切りのリストを指定します。

### 構文

**dbsrv12 -xa** [ **auth=***auth-string* [, *auth-string* ... ] ] ;**DBN=**{ * | *database-name* [, *database-name* ... ] }

### 適用対象

すべてのオペレーティングシステム、ネットワークサーバーのみ

### 備考

このオプションは、データベースミラーリングシステムで監視サーバーを起動する場合のみ指定します。

認証文字列の省略は、ミラーサーバーが提供する認証文字列の検証が行われないことを意味します。

1 つの認証文字列のみを指定した場合、すべてのデータベースがその文字列を使用し、サーバーは 1 つだけのデータベースミラーリングシステムの監視サーバーとして動作する必要があります。

複数の認証文字列が提供されている場合は、認証文字列のリストとデータベース名のリストに、同じ数のエントリーが含まれている必要があります。

指定する認証文字列は、プライマリサーバーとミラーサーバーで指定された認証文字列と一致している必要があります。

**DBN=*** の指定は、データベースがサーバーを監視サーバーとして使用できることを意味します。

## 参照

- 「CREATE MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』
- 「SET MIRROR OPTION 文」『SQL Anywhere サーバー SQL リファレンス』
- 「チュートリアル : データベースミラーリングの使用」1146 ページ
- 「チュートリアル : 監視サーバーを共有する複数のデータベースでデータベースミラーリングを使用する」1154 ページ
- 「DatabaseName (DBN) 接続パラメーター」336 ページ
- 「-xf dbsrv12 サーバーオプション」292 ページ
- 「-xp dbsrv12 データベースオプション」318 ページ

## 例

次の SQL 文は、arbiter という監視サーバーを定義します。

```
CREATE MIRROR SERVER arbiter
AS ARBITER
connection_string='SERVER=arbiter;HOST=localhost:6870';
```

次のコマンドは、監視データベースサーバーを起動し、データベースがサーバーを監視サーバーとして使用できるよう指定します。

```
dbsrv12
-n arbiter
-su sql
-x tcpip(port=6870)
-xf c:¥arbiter¥arbiterstate.txt
-xa "AUTH=abc;DBN=*"
```

次のコマンドは、監視データベースサーバーを起動し、2 つのデータベースがサーバーを監視サーバーとして使用できるよう指定します。

```
dbsrv12
-n arbiter
-su sql
-x tcpip(port=6870)
-xf c:¥arbiter¥arbiterstate.txt
-xa "AUTH=abc,xyz;DBN=database1,database2"
```

## -xd dbeng12/dbsrv12 サーバーオプション

データベースサーバーがデフォルトのデータベースサーバーにならないようにします。

### 構文

**{ dbeng12 | dbsrv12 } -xd** ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

データベースサーバーは起動するとき、そのコンピューターのデフォルトのデータベースサーバーになろうとします。デフォルトのサーバーがない場合、最初に起動したデータベースサーバーが、デフォルトのデータベースサーバーになります。そのコンピューターで、データベースサーバー名を明示的に指定しないで共有メモリに接続しようとすると、デフォルトのサーバーに接続されます。

このオプションを指定すると、データベースサーバーがデフォルトのデータベースサーバーになりません。このオプションを指定した場合、データベースサーバー名を指定しなかったクライアントは、共有メモリ経由でこのデータベースサーバーを検出できません。また、-xd オプションを指定すると、データベースサーバーでデフォルトの TCP ポートが使用されません。TCP ポートが指定されなかった場合、データベースサーバーではポート 2638 以外のポートが使用されます。

### 参照


- 「-n dbeng12/dbsrv12 サーバーオプション」257 ページ
- 「StartLine (START) 接続パラメーター」366 ページ
- 「-x dbeng12/dbsrv12 サーバーオプション」289 ページ
- 「ServerName (Server) 接続パラメーター」365 ページ
- 「データベースサーバーの検出」80 ページ


## -xf dbsrv12 サーバーオプション

データベースミラーリングシステムに関するステータス情報の管理に使用されるファイルのロケーションを指定します。このオプションは、データベースミラーリングシステムで監視サーバーを起動する場合のみコマンドで使用します。

### 構文

**dbsrv12 -xf** *state-file* ...

### 適用対象

すべてのオペレーティングシステム、ネットワークサーバーのみ

### 備考

CREATE MIRROR SERVER を使用して、パートナーサーバーのステータス情報ファイルのロケーションを定義します。

-xf オプションでは、データベースミラーリングシステムに関するステータス情報の管理に使用されるファイルのロケーションを指定します。データベースミラーリングには、このオプションは必須です。

データベースミラーリングに関するステータス情報ファイルの詳細については、「ステータス情報ファイル」1145 ページを参照してください。

### 参照

- 「CREATE MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』
- 「チュートリアル : データベースミラーリングの使用」1146 ページ
- 「チュートリアル : 監視サーバーを共有する複数のデータベースでデータベースミラーリングを使用する」1154 ページ
- 「-xa dbsrv12 サーバーオプション」290 ページ
- 「-xp dbsrv12 データベースオプション」318 ページ

### 例

次のコマンドは (全体を 1 行に入力)、ステータス情報ファイル *c:¥arbiter¥arbiter.state* を使用するデータベースサーバー myarbiter を起動します。

```
dbsrv12 -n myarbiter -su sql
-x "TCPIP(PORT=6870;DOBROAD=no)" -xf "c:¥arbiter¥arbiter.state"
-xa "AUTH=abc;DBN=mirror_demo"
```

## -xm dbeng12/dbsrv12 サーバーオプション

データベースサーバーが新しい IP アドレスをチェックする頻度を指定します。

### 構文

{ **dbeng12** | **dbsrv12** } **-xm** *seconds*

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

「クラウドに関する注意点」このオプションは、クラウドのサーバーでは完全にテストされていません。

### 備考

データベースサーバーが稼働しているコンピューターが新しいネットワークに接続され、-xm オプションが指定されている場合は、変更が検出され、データベースサーバーは新しいコンピューターで接続の受信を開始します。また、コンピューターがネットワークから切断されると、データベースサーバーはそのネットワークでの受信を停止します。

SQL Anywhere アプリケーションがポータブルデバイスで実行されていない場合、-xm オプションはデフォルトで無効になります。ポータブルデバイスでのデフォルト設定は 120 秒です。-xm オプションを無効にするには、0 を指定します。SQL Anywhere アプリケーションがポータブルデバイスで実行されているかどうかを確認するには、IsPortableDevice プロパティを使用します。

このオプションを指定しても、HTTP リスナーまたは HTTPS リスナーのパフォーマンスには影響を与えません。

MyIP (ME) ネットワークプロトコルオプションを指定した場合、-xm オプションを指定したモニタリングは無効になります。

ネットワークインターフェイスが切断されると、ネットワークインターフェイスに関連するリスナーはすべて停止します。

データベースサーバーを停止せずにこのオプションの設定を変更するには、sa_server_option システムプロシージャーを使用します。

### 参照

- IsPortableDevice サーバープロパティ 762 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「MyIP (ME) プロトコルオプション」394 ページ
- IPAddressMonitorPeriod サーバープロパティ 761 ページ

### 例

次のコマンドは、新しい IP アドレスを 60 秒ごとにチェックするデータベースサーバーを起動します。

```
dbsrv12 -xm 60;
```

## -xs dbeng12/dbsrv12 サーバーオプション

サーバー側の Web サービス通信プロトコルを指定します。

### 構文

**{ dbeng12 | dbsrv12 } -xs** { *protocol*[,...] } ...

*protocol* : {
**NONE**
| **HTTP** [ ( *option*=*value*;...) ]
| **HTTPS** [ ( *option*=*value*;...) ]

「クラウドに関する注意：」-xs オプションは、OnDemand Edition ライセンスによるクラウドサーバーで複数回指定することができます。

### 指定可能な値

次のいずれかを指定できます。

- ***option*** 各プロトコルでサポートされている *option* の値のリストについては、「ネットワークプロトコルオプション」369 ページを参照してください。

- **HTTP** HTTP プロトコルを使用するクライアントによる Web 要求を受信します。受信するデフォルトのポートは 80 です。

- **HTTPS** HTTPS プロトコルを使用するクライアントによる Web 要求を受信します。受信するデフォルトのポートは 443 です。HTTPS を使用するためには、サーバーの証明書とパスワードを指定する必要があります。HTTPS は RSA 暗号化を使用するため、パスワードは RSA 証明書であることが必要です。

  HTTP サーバーは、SSL バージョン 3.0 と TLS バージョン 1.0 を使用した HTTPS 接続をサポートしています。

  **HTTPS** を指定するか、FIPS 認定の RSA 暗号化の場合は **FIPS=Y** を付けて **HTTPS** を指定します。FIPS 認定の HTTPS は別の認定ライブラリを使用しますが、HTTPS と互換性があります。

  FIPS 認定のアルゴリズムの実行については、「-fips dbeng12/dbsrv12 サーバーオプション」230 ページを参照してください。

- **NONE** Web 要求を受信しません。これはデフォルトです。

使用可能なパラメーターの詳細については、「ネットワークプロトコルオプション」369 ページを参照してください。

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

-xs オプションを使用して、要求の受信に使用する Web プロトコルを指定します。

-xs オプションを指定しない場合、データベースサーバーは Web 要求を受信しようとしません。

-xs オプションとともに 1 つ以上のプロトコルを指定すると、サーバーは、指定したプロトコルを使用して、Web 要求を受信しようとします。

> **注意**
> 複数の Web サーバーを同時に起動する場合、どちらも同じデフォルトポートを使用するため、どちらか 1 つのポートを変更する必要があります。

トランスポートレイヤーセキュリティには、HTTPS または FIPS 認定の HTTPS プロトコルを使用できます。「SQL Anywhere Web サービスの暗号化」1425 ページを参照してください。

**注意**
別途ライセンスが必要な必須コンポーネント

ECC 暗号化と FIPS 認定の暗号化には、別途ライセンスが必要です。強力な暗号化テクノロジはすべて、輸出規制対象品目です。

「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

UNIX では、複数のパラメーターを指定する場合に二重引用符が必要です。

```
-xs "HTTP(OPTION1=value1;OPTION2=value2;...)"
```

### 参照


- 「-sn dbsrv12 オプション」315 ページ
- 「-x dbeng12/dbsrv12 サーバーオプション」289 ページ
- 「-xa dbsrv12 サーバーオプション」290 ページ
- 「-xf dbsrv12 サーバーオプション」292 ページ
- 「-xp dbsrv12 データベースオプション」318 ページ
- 「FIPS プロトコルオプション」385 ページ
- 「HTTP Web サーバーとしての SQL Anywhere」『SQL Anywhere サーバー プログラミング』


### 例

HTTP Web 要求をポート 80 で受信します。

```
dbeng12 web.db -xs HTTP(PORT=80)
```

HTTPS を使用して Web 要求を受信します。

```
dbeng12 web.db -xs HTTPS(FIPS=N;PORT=82;IDENTITY=eccserver.id;IDENTITY_PASSWORD=test)
```

## -z dbeng12/dbsrv12 サーバーオプション

診断通信メッセージなどのメッセージをトラブルシューティングのために表示します。

### 構文

**{ dbeng12 | dbsrv12 } -z** ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

このオプションは、問題の原因を突き止める場合にだけ使用します。情報は、データベースサーバーメッセージウィンドウに表示されます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照

- DebuggingInformation サーバープロパティ758 ページ
- 「-ze dbeng12/dbsrv12 オプション」297 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』

## -ze dbeng12/dbsrv12 オプション

データベースサーバー環境変数をデータベースサーバーメッセージウィンドウに表示します。

### 構文

**{ dbeng12 | dbsrv12 } -ze** ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー (Windows Mobile を除く)

### 備考

-ze オプションを指定すると、起動時に環境変数がデータベースサーバーメッセージウィンドウに表示されます。データベースサーバーメッセージウィンドウの内容をファイルに保存するには、データベースサーバーの起動時に -o オプションを指定します。

### 参照

- 「SQL Anywhere の環境変数」443 ページ
- 「-o dbeng12/dbsrv12 サーバーオプション」258 ページ
- 「-z dbeng12/dbsrv12 サーバーオプション」296 ページ

### 例

次のコマンドは、データベースサーバー myserver を起動し、このサーバーに設定されている環境変数をデータベースサーバーメッセージウィンドウと *server-log.txt* ファイルに出力します。

```
dbeng12 -n myserver -ze -o server-log.txt
```

## -zl dbeng12/dbsrv12 サーバーオプション

サーバー上の各データベース接続の、最後に作成された SQL 文の取得をオンにします。

### 構文

**{ dbeng12 | dbsrv12 } -zl** ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

この機能は、RememberLastStatement サーバー設定を使用してオンにすることもできます。ある接続で最後に準備された SQL 文は、CONNECTION_PROPERTY 関数の LastStatement 値を使用し

て取得できます。sa_conn_activity ストアドプロシージャーを使用すると、サーバー上の現在のすべてのデータベース接続について、最後に作成された SQL 文を取得できます。

LastStatement の値は、文が準備されると同時に設定され、文が削除されると同時にクリアされます。各接続につき 1 つの文の文字列のみが記憶されます。

ある接続について sa_conn_activity が空でない値を返した場合、その接続で現在実行されている文である可能性が高くなります。その文が完了している場合は、文がすでに削除され、このプロパティの値がクリアされている可能性があります。アプリケーションが複数の文を準備し、それらの文のステートメントハンドルを保持している場合、LastStatement が返す値は接続で現在実行されている処理を表しません。

ストアドプロシージャーコールの場合、プロシージャー内の文ではなく、最も外側のプロシージャーコールのみが表示されます。

**警告**
-zl が指定されている場合、または RememberLastStatement サーバーの設定がオンになっている場合は、任意のユーザーが sa_conn_activity システムプロシージャーを呼び出すか LastStatement 接続プロパティの値を取得して、他のユーザーに対して最後に作成された SQL 文を調べることができます。このオプションは注意して使用し、不要な場合はオフにしてください。

### 参照


- RememberLastStatement サーバープロパティ 773 ページ
- LastStatement 接続プロパティ 728 ページ
- 「sa_conn_activity システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』


## -zn dbeng12/dbsrv12 サーバーオプション

保持する要求ログファイルのコピー数を指定します。

### 構文

**{ dbeng12 | dbsrv12 } -zn** *integer*

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

要求ロギングが長時間にわたって有効になっていると、要求ログファイルが大きくなることがあります。-zn オプションを使用すると、保持する要求ログファイルのコピー数を指定できます。ただし、-zs も指定されていなければ有効になりません。-zs オプションにより、元のログファイルが指定のサイズに到達すると、新しいログファイルを作成し、元のログファイルの名前を変更できます。

たとえば、要求ロギング情報を *req.out* ファイルにリダイレクトし、-zn オプションを使って 5 つのログファイルコピーを要求すると、サーバーは *req.out.1*、*req.out.2*、*req.out.3*、*req.out.4*、*req.out.*

*5* の順にファイルを作成します。これらのファイルが存在する場合、アクティブな要求ログが再び一杯になると以下の動作が発生します。

●*req.out.1* が削除される。

●*req.out.2* ～ *req.out.5* のファイル名が *req.out.1* ～ *req.out.4* に変更される。

●アクティブなログファイルのコピーの名前が *req.out.5* に変更される。

要求ロギングを有効にするには、-zr オプションを使用します。このログは、-zo オプションにより別のファイルにリダイレクトできます。また、sa_server_option システムプロシージャーを使用して、要求ログの数を設定することもできます。その場合、*nn* には、要求ログファイルのコピーの数を指定します。

```
CALL sa_server_option('RequestLogNumFiles',nn);
```

### 参照


- 「-zs dbeng12/dbsrv12 サーバーオプション」303 ページ
- RequestLogNumFiles サーバープロパティ775 ページ
- 「-zo dbeng12/dbsrv12 サーバーオプション」299 ページ
- 「-zr dbeng12/dbsrv12 サーバーオプション」301 ページ
- 「-zs dbeng12/dbsrv12 サーバーオプション」303 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「要求ロギング」『SQL Anywhere サーバー SQL の使用法』


### 例

次の例 (全体を 1 行に入力) では、要求ロギング情報は *mydatabase.log* という要求ログファイルに出力されます。このファイルは最大サイズが 10 KB で、要求ログのコピーが 3 つ保持されます。

```
dbeng12 "c:¥my data¥mydatabase.db" -zr all -zn 3
 -zs 10 -zo mydatabase.log
```

## -zo dbeng12/dbsrv12 サーバーオプション

通常のログファイルとは別のファイルに、要求ロギング情報をリダイレクトします。

### 構文

{ **dbeng12** | **dbsrv12** } **-zo** *filename* ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

要求ロギングは、-zr オプションを使用すると有効になります。-zo オプションを指定することによって、出力をこのファイルから通常のログファイルではない別のファイルにリダイレクトできます。

また、このオプションにより、要求ロギングがデータベースサーバーメッセージウィンドウに表示されなくなります。

### 参照

- RequestLogFile サーバープロパティ 774 ページ
- 「-zn dbeng12/dbsrv12 サーバーオプション」298 ページ
- 「-zr dbeng12/dbsrv12 サーバーオプション」301 ページ
- 「-zs dbeng12/dbsrv12 サーバーオプション」303 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「要求ロギング」『SQL Anywhere サーバー SQL の使用法』

## -zoc dbeng12/dbsrv12 サーバーオプション

Web サービスクライアント情報をファイルにリダイレクトします。

### 構文

{ **dbeng12** | **dbsrv12** } **-zoc** *filename* ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

Web サービスクライアントログファイルには、HTTP 要求とアウトバウンド Web サービスクライアントログコール用に記録されたトランスポートデータが含まれています。-zoc サーバーオプションを指定すると、ロギングが自動的に有効になります。このファイルへのロギングの有効／無効を切り替えるには、sa_server_option システムプロシージャーを使用します。

```
CALL sa_server_option( 'WebClientLogging', 'ON' );
```

### 参照

- WebClientLogFile サーバープロパティ 781 ページ
- WebClientLogging サーバープロパティ 781 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「HTTP Web サーバーとしての SQL Anywhere」『SQL Anywhere サーバー プログラミング』
- 「CREATE FUNCTION 文 [Web サービス]」『SQL Anywhere サーバー SQL リファレンス』
- 「CREATE PROCEDURE 文 [Web サービス]」『SQL Anywhere サーバー SQL リファレンス』

### 例

次のコマンドは、データベースサーバーを起動し、HTTP Web 要求をポート 80 で受信し、アウトバウンド Web サービスクライアント情報を *clientinfo.txt* ファイルに記録します。

```
dbeng12 web.db -xs HTTP(PORT=80) -zoc clientinfo.txt
```

## -zp dbeng12/dbsrv12 サーバーオプション

クエリオプティマイザーが最近使用したプランの取得をオンにします。

### 構文

**{ dbeng12 | dbsrv12 } -zp** ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

このオプションは、各接続で最近使用されたクエリ実行プランをデータベースサーバーに保存する場合に使用します。sa_server_option システムプロシージャーの RememberLastPlan サーバー設定を使用して、この機能をオンにすることもできます。LastPlanText 接続プロパティを使用することによって、最近使用されたプランのテキストを表示できます。

### 参照


- RememberLastPlan サーバープロパティ 773 ページ
- LastPlanText 接続プロパティ 727 ページ
- 「sa_conn_activity システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』


## -zr dbeng12/dbsrv12 サーバーオプション

操作の要求ロギングを有効にします。

### 構文

**{ dbeng12 | dbsrv12 } -zr { SQL | HOSTVARS | PLAN | PROCEDURES | TRIGGERS | OTHER | BLOCKS | REPLACE | ALL | NONE }** ...

### 指定可能な値

- **SQL**　以下の項目のロギングを有効にします。

  - START DATABASE 文
  - STOP DATABASE 文
  - STOP SERVER 文
  - 文の準備と実行
  - EXECUTE IMMEDIATE 文
  - オプション設定
  - COMMIT 文
  - ROLLBACK 文
  - PREPARE TO COMMIT 操作
  - 接続と接続解除
  - トランザクションの開始
  - DROP STATEMENT 文
  - カーソルの説明
  - カーソルを開く、閉じる、再開する
  - エラー

- **PLAN**　実行プランのロギングを有効にします (短いプラン)。プロシージャーの実行プランは、プロシージャー (PROCEDURES) のロギングが有効な場合にも記録されます。

- **HOSTVARS**　ホスト変数の値のロギングを有効にします。HOSTVARS を指定した場合、SQL にリストされている情報もロギングされます。

- **PROCEDURES**　プロシージャー内から実行されている文のロギングを有効にします。

- **TRIGGERS**　トリガー内から実行されている文のロギングを有効にします。

- **OTHER**　SQL に含まれないその他の要求タイプのロギングを有効にします (FETCH や PREFETCH など)。ただし、OTHER を指定して SQL を指定しない場合、SQL+OTHER を指定した場合と同じです。OTHRE を含めると、ログファイルが急速に拡大し、サーバーのパフォーマンス低下につながることがあります。

- **BLOCKS**　別の接続で接続がブロックされたときと、接続のブロックが解除されたときに表示する詳細のロギングを有効にします。

- **REPLACE**　ロギングの開始時に、既存の要求ログは同じ名前を持つ新規の (空の) ログで置換されます。それ以外の場合、既存の要求ログが開き、新規エントリがファイルの末尾に追加されます。

- **ALL**　すべてのサポート情報をロギングします。この設定は、SQL+PLAN+HOSTVARS +PROCEDURES+TRIGGERS+OTHER+BLOCKS を指定した場合と同じです。この設定では、ログファイルが急速に拡大し、サーバーのパフォーマンス低下につながることがあります。

- **NO または NONE**　要求ログに対するロギングを無効にします。

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 備考

このオプションは、問題の原因を突き止める場合にだけ使用します。情報は、データベースサーバーメッセージウィンドウに表示されるか、要求ログに送信されます。複数の値を指定する場合、これらの値は , または + で区切られます。

データベースサーバーを起動した後で、sa_server_option システムプロシージャーを使用し、要求ログ設定を変更してロギングの対象とする情報を増減できます。

RequestLogging 設定の現在の値は、次のクエリを使用して検索できます。

```
SELECT PROPERTY( 'RequestLogging' );
```

### 参照


- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- RequestLogging サーバープロパティ 775 ページ
- 「-zn dbeng12/dbsrv12 サーバーオプション」 298 ページ
- 「-zo dbeng12/dbsrv12 サーバーオプション」 299 ページ
- 「-zs dbeng12/dbsrv12 サーバーオプション」 303 ページ
- 「要求ロギング」『SQL Anywhere サーバー SQL の使用法』


## -zs dbeng12/dbsrv12 サーバーオプション

要求ログのサイズを制限します。

### 構文

**{ dbeng12 | dbsrv12 } -zs** { *size*[ **k** | **m** | **g** ] } ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

要求ロギングを有効にするには、-zr オプションを使用します。このログは、-zo オプションにより別のファイルにリダイレクトできます。また、-zs オプションを使用してファイルのサイズを制限できます。

*size* には、要求ログの最大ファイルサイズを、バイト単位で指定します。単位をキロバイト、メガバイト、またはギガバイトで指定するには、それぞれ **k**、**m**、**g** のいずれかを使用してください。

-zs 0 を指定した場合は、要求ロギングファイルの最大サイズは適用されず、名前は変更されません。これはデフォルト値です。

要求ログファイルが -zs オプションまたは sa_server_option システムプロシージャーで指定したサイズに到達すると、ファイルが拡張子 *.old* の付いた名前に変更されます (既存のファイルが存在する場合は、同じ名前で置換されます)。要求ログファイルが再起動します。

### 参照

- RequestLogMaxSize サーバープロパティ 775 ページ
- 「-zn dbeng12/dbsrv12 サーバーオプション」 298 ページ
- 「-zo dbeng12/dbsrv12 サーバーオプション」 299 ページ
- 「-zr dbeng12/dbsrv12 サーバーオプション」 301 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「要求ロギング」『SQL Anywhere サーバー SQL の使用法』

### 例

次の例では、ログファイルのサイズを制御するための -zs オプションの使用方法を示します。次のコマンドラインを使用してデータベースサーバーを起動するとします。

```
dbeng12 -zr all -zs 10k -zo mydatabase.log
```

新規ログファイル *mydatabase.log* が作成されます。このファイルのサイズが 10 KB に達すると、既存の *mydatabase.old* ファイルが削除され、*mydatabase.log* の名前が *mydatabase.old* に変更されて、新しい *mydatabase.log* ファイルが開始されます。このプロセスは、*mydatabase.log* ファイルが指定したサイズ (この場合は 10 KB) に達するたびに繰り返されます。

## -zt dbeng12/dbsrv12 オプション

要求タイミング情報のロギングをオンにします。

### 構文

**{ dbeng12 | dbsrv12 } -zt** ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

-zt オプションを指定すると、RequestTiming 接続プロパティが On に設定されます。要求のタイミングがオンになると、sa_performance_diagnostics システムプロシージャーを使用して要求のタイミング情報を格納する接続プロパティのすべての値を取得できます。

次の接続プロパティは、RequestTiming がオンになったときにのみ、要求タイミング情報を返します。それ以外の場合、プロパティは NULL を返します。

- ReqCountActive 接続プロパティ
- ReqCountBlockContention 接続プロパティ
- ReqCountBlockLock 接続プロパティ
- ReqCountUnscheduled 接続プロパティ
- ReqStatus 接続プロパティ
- ReqTimeActive 接続プロパティ
- ReqTimeBlockContention 接続プロパティ
- ReqTimeBlockLock 接続プロパティ
- ReqTimeUnscheduled 接続プロパティ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

RequestTiming 設定の現在の値は、次のクエリを使用して検索できます。

```
SELECT PROPERTY( 'RequestTiming' );
```

### 参照

- RequestTiming サーバープロパティ 775 ページ
- 「sa_performance_diagnostics システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「sa_performance_statistics システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「要求ロギング」『SQL Anywhere サーバー SQL の使用法』

# データベースオプション (dbeng12/dbsrv12)

次のオプションは、データベースファイルの後に指定し、そのデータベースだけに適用されます。

## -a dbeng12/dbsrv12 データベースオプション

指定したトランザクションログを適用します。

### 構文

{ **dbeng12** | **dbsrv12** } [ *server-options* ] *database-file* **-a** *log-filename* ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

このオプションは、データベースファイルのメディア障害からのリカバリに使用されます。このオプションを指定すると、データベースサーバーはログを適用して終了します (実行は継続しません)。複数のトランザクションログを適用する場合は、-a オプションを使用するときにトランザクションログの正しい適用順序を把握する必要があります。-ad オプションまたは -ar オプションを使用すると、複数のトランザクションログが自動的に正しい順序で適用されます。

-a データベースオプションは、*database-file* の後に指定する必要があり、そのデータベースだけに適用されます。

サーバーの起動時にキャッシュサイズを指定すると、リカバリ時間を短縮できます。

「バックアップとデータリカバリ」1071 ページを参照してください。

### 参照


- 「データのメディア障害からのリカバリ」1098 ページ
- 「トランザクションログが複数の場合のデータベースのリカバリ」1094 ページ
- 「-ad dbeng12/dbsrv12 データベースオプション」306 ページ
- 「-ar dbeng12/dbsrv12 データベースオプション」307 ページ
- 「-as dbeng12/dbsrv12 データベースオプション」308 ページ


### 例

次の例 (全体を 1 行に入力) では、ログファイル *demo.log* がサンプルデータベースのバックアップコピーに適用されます。

```
dbeng12 "c:¥backup¥demo.db" -a "c:¥backup¥demo.log"
```

## -ad dbeng12/dbsrv12 データベースオプション

データベースに適用されるトランザクションログファイルがあるディレクトリを指定します。

### 構文

{ **dbeng12** | **dbsrv12** } [ *server-options* ] *database-file* **-ad** *log-directory* ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

-ad オプションを使用すると、指定したディレクトリでデータベースのトランザクションログファイルが検索されます。トランザクションログファイルの開始ログオフセットがデータベースファイルに保存されている開始ログオフセットと等しいか、それより大きい場合、トランザクションログファイルはログオフセット順に適用されます。すべてのトランザクションログファイルが適用されると、データベースは停止します。トランザクションログファイルの適用が完了した後もデータベースの実行を継続する場合は、-as オプションも一緒に指定する必要があります。

指定したログディレクトリが相対パスを使用する場合、このディレクトリは *database-file* ディレクトリに対して相対的に読まれます。

-ad データベースオプションは、*database-file* の後に指定する必要があり、そのデータベースだけに適用されます。

### 参照


- 「データのメディア障害からのリカバリ」1098 ページ
- 「トランザクションログが複数の場合のデータベースのリカバリ」1094 ページ
- 「-a dbeng12/dbsrv12 データベースオプション」305 ページ
- 「-ar dbeng12/dbsrv12 データベースオプション」307 ページ
- 「-as dbeng12/dbsrv12 データベースオプション」308 ページ


 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 例

次の例では、*backup* ディレクトリ内のログファイルが *mysample.db* データベースに適用され、適用が完了すると、データベースが停止します。

```
dbeng12 "c:¥mysample.db" -ad "c:¥backup"
```

次の例では、*backup* ディレクトリ内のログファイルが *mysample.db* データベースに適用され、適用の完了後もデータベースの実行は継続します。

```
dbeng12 "c:¥mysample.db" -ad "c:¥backup" -as
```

## -ar dbeng12/dbsrv12 データベースオプション

現在のトランザクションログと同じディレクトリ内にあるトランザクションログファイルをデータベースに適用します。

### 構文

{ **dbeng12** | **dbsrv12** } [ *server-options* ] *database-file* **-ar** ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

-ar オプションを使用すると、データベースサーバーのトランザクションログファイルは現在のトランザクションログと同じディレクトリで検索されます。トランザクションログのディレクトリはデータベースから取得されます。トランザクションログファイルの開始ログオフセットがデータベースに保存されている開始ログオフセットと等しいか、それより大きい場合、トランザクションログファイルはログオフセット順に適用されます。すべてのトランザクションログファイルが適用されると、データベースは停止します。トランザクションログファイルの適用が完了した後もデータベースの実行を継続する場合は、-as オプションも一緒に指定する必要があります。

-ar データベースオプションは、*database-file* の後に指定する必要があり、そのデータベースだけに適用されます。

### 参照


- 「データのメディア障害からのリカバリ」1098 ページ
- 「トランザクションログが複数の場合のデータベースのリカバリ」1094 ページ
- 「-a dbeng12/dbsrv12 データベースオプション」305 ページ
- 「-ad dbeng12/dbsrv12 データベースオプション」306 ページ
- 「-as dbeng12/dbsrv12 データベースオプション」308 ページ


### 例

次の例では、トランザクションログファイル (保存されているディレクトリはデータベースから取得されます) が *mysample.db* データベースに適用されます。トランザクションログファイルの適用が完了した後もデータベースの実行は継続します。

```
dbeng12 "c:¥mysample.db" -ar -as
```

## -as dbeng12/dbsrv12 データベースオプション

トランザクションログの適用後もデータベースの実行を継続します (-ad または -ar とともに使用)。

### 構文

{ **dbeng12** | **dbsrv12** } [ *server-options* ] *database-file* { **-ad** *log-dir* | **-ar** } **-as** ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

-as オプションを指定する場合は、-ad オプションまたは -ar オプションも指定する必要があります。-as オプションを指定すると、トランザクションログの適用後もデータベースの実行が継続します。

-as データベースオプションは、*database-file* の後に指定する必要があり、そのデータベースだけに適用されます。

### 参照


- 「データのメディア障害からのリカバリ」1098 ページ
- 「トランザクションログが複数の場合のデータベースのリカバリ」1094 ページ
- 「-a dbeng12/dbsrv12 データベースオプション」305 ページ
- 「-ad dbeng12/dbsrv12 データベースオプション」306 ページ
- 「-ar dbeng12/dbsrv12 データベースオプション」307 ページ


### 例

次の例では、トランザクションログファイルが *mysample.db* データベースに適用されます。-ar が指定されているため、データベースサーバーはトランザクションログのロケーションをデータベースから取得します。ログファイルの適用が完了した後もデータベースの実行は継続します。

```
dbeng12 "c:¥mysample.db" -ar -as
```

次の例では、*backup* ディレクトリ内のログファイルが *mysample.db* データベースに適用されます。ログファイルの適用が完了した後もデータベースの実行は継続します。

```
dbeng12 "c:¥mysample.db" -ad "c:¥backup" -as
```

## -ds dbeng12/dbsrv12 データベースオプション

データベースの DB 領域とトランザクションログが配置されているディレクトリを指定します。

### 構文

{ **dbeng12** | **dbsrv12** } *database-file* **-ds** *dbspace-directory* ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

DB 領域のディレクトリを指定すると、データベースサーバーは、DB 領域を探すときにこのディレクトリだけを検索対象とします。DB 領域のロケーションは、データベースサーバーメッセージウィンドウに表示されます。

フルパス名を指定された DB 領域がバックアップに含まれている場合は、このオプションを使用して、元のデータベースが実行中である間に、元のデータベースと同じコンピューター上にあるデータベースのバックアップコピーを起動できます。

-ds データベースオプションは、*database-file* の後に指定する必要があり、そのデータベースだけに適用されます。

このオプションで指定されたディレクトリでトランザクションログが見つからない場合は、このロケーションにトランザクションログが作成されます。

> **警告**
> -ds オプションはリカバリにのみ使用してください。このオプションを指定し、ここで指定したディレクトリに現在のライブトランザクションログが存在しない場合は、新しいトランザクションログが指定のロケーションで作成されます。

### 参照


- 「追加の DB 領域の考慮事項」 121 ページ
- 「START DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「STOP DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「default_dbspace オプション」 617 ページ


### 例

次の例では、ディレクトリ *c:¥backup¥Nov15* に対して DB 領域を検索するデータベースサーバーを起動します。

```
dbeng12 c:¥backup¥Nov15¥my.db -ds c:¥backup¥Nov15¥
```

次の例では、現在のディレクトリに対して DB 領域を検索するデータベースサーバーを起動します。

```
dbeng12 my.db -ds .
```

## -dh dbeng12/dbsrv12 データベースオプション

このオプションを指定すると、サーバーに対してサーバー列挙ユーティリティ (dblocate) を実行した場合、データベースが表示されません。

### 構文

{ **dbeng12** | **dbsrv12** } [ *server-options* ] *database-file* **-dh** ...

### 適用対象

すべてのプラットフォーム

### 備考

-dh オプションを指定すると、サーバーに対してサーバー列挙ユーティリティ (dblocate) を実行したときに、データベースが検出されません。そのため、-d、-dn、-dv のいずれかのオプションを指定して dblocate を実行した場合、-dh オプションを指定したデータベースは表示されません。

-dh データベースオプションは、*database-file* の後に指定する必要があり、そのデータベースだけに適用されます。

### 参照

- 「サーバー列挙ユーティリティ (dblocate)」999 ページ

## -ek dbeng12/dbsrv12 データベースオプション

強力に暗号化されたデータベースのキーを指定します。

### 構文

{ **dbeng12** | **dbsrv12** } [ *server-options* ] *database-file* **-ek** *key* ...

### 適用対象

すべてのオペレーティングシステムとサーバー。

### 備考

暗号化されたデータベースを起動するには、key 値を -ek オプションに指定してください。key は、大文字、小文字、数字、文字、特殊記号を含む文字列です。

クリアテキストで見ることができないように暗号化キーをウィンドウに入力するには、-ep サーバーオプションを使用します。

クライアントアプリケーションとデータベースサーバー間の通信パケットを安全化するには、-ec サーバーオプションとトランスポートレイヤーセキュリティを使用します。「トランスポートレイヤーセキュリティ」1410 ページを参照してください。

-ek データベースオプションは、*database-file* の後に指定する必要があり、そのデータベースだけに適用されます。

### 参照

- 「-ec dbeng12/dbsrv12 サーバーオプション」223 ページ
- 「-ep dbeng12/dbsrv12 サーバーオプション」226 ページ
- 「DatabaseKey (DBKEY) 接続パラメーター」335 ページ
- 「データベースの暗号化と復号化」1396 ページ

### 例

次の例では、データベースを起動して、コマンドラインで暗号化キーを指定します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
dbsrv12 -x tcpip mydata.db -ek "Akmm9u70y"
```

## -m dbeng12/dbsrv12 データベースオプション

チェックポイントが終了すると、トランザクションログをトランケートします。

### 構文

{ **dbeng12** | **dbsrv12** } [ *server-options* ] *database-file* **-m** ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

シャットダウン時、またはサーバーでスケジュールされたチェックポイントの結果としてチェックポイントが実行されたときに、トランザクションログをトランケートします。これでトランザクションログの肥大化が自動的に制限されます。チェックポイントの頻度は、checkpoint_time と recovery_time オプションによって制御できます (または -gc および -gr dbeng12/dbsrv12 サーバーオプションでも設定できます)。

-m オプションは、高速な応答時間を必要とする大容量のトランザクションを処理する場合や、リカバリやレプリケーションがトランザクションログの内容に依存しない場合に役立ちます。このオプションを選択すると、データベースファイルを含むデバイスのメディア障害に対して無防備な状態になります。

データベースファイルの断片化を防ぐためには、このオプションを使用する場合に、トランザクションログをデータベースそのものとは別のデバイスまたはパーティションに保管することをおすすめします。

このオプションは -m サーバーオプションと同じですが、現在のデータベースまたは *database-file* 変数で識別されるデータベースにのみ適用されます。

**警告**
レプリケートされるデータベースまたは同期されるデータベースでは、-m オプションを使わないでください。SQL Remote と Mobile Link で使用されるレプリケーションと同期は、本質的にトランザクションログ情報に依存します。

-m データベースオプションは、*database-file* の後に指定する必要があり、そのデータベースだけに適用されます。

### 参照


- 「-m dbeng12/dbsrv12 サーバーオプション」255 ページ
- 「トランザクションログ」127 ページ
- 「トランザクションログユーティリティ (dblog)」1041 ページ

### 例

次の例では、silver という名前のデータベースサーバーが起動され、データベース *salesdata.db* がロードされます。チェックポイントが終了すると、トランザクションログの内容が削除されます。

```
dbsrv12 -n silver "c:¥inventory details¥salesdata.db" -m
```

## -n dbeng12/dbsrv12 データベースオプション

データベースの名前を設定します。

### 構文

{ **dbeng12** | **dbsrv12** } [ *server-options* ] *database-file* **-n** *string* ...

### デフォルト

データベースには、パスと拡張子を削除したデータベースファイル名が付けられます。

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

データベースサーバーとデータベースはどちらも名前を付けることができます。データベースサーバーはいくつかのデータベースをロードできるので、データベース名を使用して、各データベースを区別します。

データベース名には、次に該当する値を指定できません。

- 空白スペース、一重引用符、または二重引用符で始まる値
- 空白スペースで終わる値
- セミコロンを含む値
- 長さが 250 バイトを超える値

ユーティリティデータベースへの接続に使用できるのは utility_db という名前のデータベースのみです。「ユーティリティデータベース」135 ページを参照してください。

-n データベースオプションは、*database-file* の後に指定する必要があり、そのデータベースだけに適用されます。

### 参照


- 「データベースサーバー名とデータベース名を指定する」151 ページ
- 「-n dbeng12/dbsrv12 サーバーオプション」257 ページ


### 例

*%SQLANYSAMP12%¥demo.db* ファイルでデータベースを起動するときに、-n オプションを指定しなかった場合、データベースの名前は demo になります。

次の例では、データベースサーバーがキャッシュサイズ 3 MB で起動され、データベースがロードされます。データベースには test という名前が付けられます。データベースサーバー名が指定されていないため、サーバーは最初のデータベースから名前を取得します。そのため、サーバーの名前も test になります。

```
dbsrv12 -c 3MB "c:¥mydata.db" -n "test"
```

**注意**
2 つの -n オプションがあります。-n オプションの内容は指定する位置によって異なります。データベースファイル名の前に指定すると、サーバーオプションとしてサーバー名を指定します。データベースファイル名の後に指定すると、データベースオプションとしてデータベース名を指定します。

たとえば、次のコマンドでは、サーバー名 SERV とデータベース名 DATA が指定されます。

```
dbsrv12 -n SERV c:¥mydata.db -n DATA
```

「-n dbeng12/dbsrv12 サーバーオプション」257 ページを参照してください。

## -r dbeng12/dbsrv12 データベースオプション

指定されたデータベースを読み込み専用として起動します。

### 構文

{ **dbeng12** | **dbsrv12** } [ *server-options* ] *database-file* **-r** ...

### 適用対象

すべてのオペレーティングシステムとデータベースサーバー

### 備考

コマンドラインでどのデータベース名よりも前にオプションを指定した場合、テンポラリファイルを除くすべてのデータベースファイル (メインデータベースファイル、DB 領域、トランザクションログ、トランザクションログミラー) が読み込み専用モードで開きます。データベースへの変更はできません。つまり、データベースサーバーはデータベースファイルやトランザクションログファイルを変更しません。

あるデータベースファイルの後ろに -r オプションを指定した場合、そのデータベースだけが読み込み専用になります。テンポラリテーブルに変更を加えることはできますが、トランザクションログとロールバックログが無効化されているため、ROLLBACK を実行しても効果はありません。

変更が不可能なデータベースファイルの例として、CD-ROM によって配布されるデータベースがあります。このようなデータベースには読み込み専用モードでアクセスできます。

INSERT 文や DELETE 文などを使ってデータベースを変更しようとすると、SQLSTATE_READ_ONLY_DATABASE エラーが発生します。

リカバリを必要とするデータベースは、読み込み専用モードでは起動できません。たとえば、オンラインバックアップを使って作成したデータベースファイルは、バックアップの開始時に開いているトランザクションがあると、読み込み専用モードで起動できません。これは、バックアップコピーの開始時に、これらのトランザクションがリカバリを必要とするためです。

監査がオンである場合、データベースを読み込み専用モードで起動することはできません。

### 参照


- 「-r dbeng12/dbsrv12 サーバーオプション」267 ページ
- 「auditing オプション」598 ページ


### 例

2 つのデータベースを読み込み専用モードで開くには、次のように実行します。

```
dbeng12 -r database1.db database2.db
```

2 つのうち最初のデータベースだけを読み込み専用モードで開くには、次のように実行します。

```
dbeng12 database1.db -r database2.db
```

## -sm dbsrv12 データベースオプション (旧式)

読み込み専用のミラーデータベースへのアクセスに使用できる代替データベースサーバー名を指定します。このオプションは推奨されなくなりました。代わりに CREATE MIRROR SERVER 文を使用します。

### 構文

**dbsrv12** [ *server-options* ] *database-file* **-sm** *alternate-server-name*

### 適用対象

すべてのオペレーティングシステム、ネットワークサーバーのみ

### 備考

*alternate-server-name* は、そのデータベースサーバーがデータベースのミラーとして動作している場合のみアクティブになります。-sm と -sn コマンドラインオプションを使用すると、どの物理サーバーがプライマリまたはミラーとして動作しているかを把握していなくても、アプリケーションは常にプライマリサーバーまたはミラーサーバーのデータベースに接続できます。

-sm データベースオプションは、*database-file* の後に指定する必要があり、そのデータベースだけに適用されます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照

- 「CREATE MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』
- 「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』
- 「ミラーサーバーで実行されているデータベースへの読み込み専用アクセスの設定」1165 ページ
- 「-xa dbsrv12 サーバーオプション」290 ページ
- 「-xf dbsrv12 サーバーオプション」292 ページ
- 「-xp dbsrv12 データベースオプション」318 ページ
- 「START DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「データベースミラーリング」1137 ページ
- 「サーバー列挙ユーティリティ (dblocate)」999 ページ
- ReadOnly データベースプロパティ803 ページ

### 例

次のコマンドは、データベースサーバー myserver 上のデータベース *satest.db* と *sample.db* を起動します。-sn オプションを指定すると、データベースサーバーは *sample.db* に接続するときに、代替サーバー名として mysampleprimary を使用します。一方、-sm オプションを指定すると、データベースサーバーはミラーサーバーで実行中の *sample.db* に接続するときに、代替サーバー名として mysamplemirror を使用します。

```
dbsrv12 -n myserver satest.db sample.db -sn mysampleprimary
-sm mysamplemirror -xp "partner=( Server=server2;LINKS=TCPIP( PORT=2637;TIMEOUT=1 ) );
auth=abc;arbiter=( Server=arbiter;LINKS=TCPIP( PORT=2639;TIMEOUT=1 ) );mode=sync"
-xf "c:¥arbiter¥arbiter.state"
```

プライマリサーバーで実行中の *sample.db* に接続するには、次のいずれかの接続パラメーターを使用します。

- Server=myserver;DBN=sample
- Server=mysampleprimary
- Server=mysampleprimary;DBN=sample

Server=mysampleprimary を使用して *satest.db* に接続することはできません。

ミラーサーバーで実行中の *sample.db* に接続するには、次のいずれかの接続パラメーターを使用します。

- Server=myserver;DBN=sample
- Server=mysamplemirror
- Server=mysamplemirror;DBN=sample

Server=mysamplemirror を使用して *satest.db* に接続することはできません。

## -sn dbsrv12 オプション

データベースサーバー上で動作する 1 つのデータベースに代替サーバー名を割り当てます。

### 構文

**dbsrv12** [ *server-options* ] *database-file* **-sn** *alternate-server-name*

### 適用対象

すべてのオペレーティングシステム、ネットワークサーバーのみ

### 備考

**注意**
データベースミラーリングを使用している場合は、プライマリサーバーとミラーサーバーがそれぞれどちらのサーバーなのかを事前に把握していなくてもクライアントアプリケーションが現在のプライマリサーバーに接続できるように、代替サーバー名を指定する必要があります。両方のパートナーサーバーでは、代替サーバー名として同じ名前を使用することが必要です。CREATE MIRROR SERVER 文を使用すると、ミラーリングシステムのプライマリサーバーとミラーサーバーの代替サーバー名を作成できます。「CREATE MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

-sn データベースオプションは、*database-file* の後に指定する必要があり、そのデータベースだけに適用されます。

データベースサーバーが特定のデータベースサーバーに対応する複数のサーバー名を受信するように設定できます。実サーバー名以外のサーバー名を代替サーバー名といいます。代替サーバー名はデータベースサーバー上の特定のデータベースに固有のものです。クライアントは、ServerName 接続パラメーターを使用して代替サーバー名を指定し、そのデータベースに接続できます。

代替サーバー名はネットワーク全体でユニークでなければなりません。データベースをコマンドラインで起動し、指定した代替サーバー名がユニークでない場合、データベースサーバーの起動は失敗します。START DATABASE 文を使用して代替サーバー名を指定することもできます。

代替サーバー名を指定するクライアントが接続できるのは、その代替サーバー名が指定されたデータベースだけです。同じデータベースサーバー他のデータベースに対して、接続、作成、停止、削除を行うことはできません。接続パラメーターの DBN または DBF を指定する場合は、それぞれデータベース名またはデータベースファイルと一致している必要があります。接続パラメーターの DBN または DBF を指定しない場合、データベースはそのサーバーのデフォルトデータベースと同じように動作します。

サーバー列挙ユーティリティ (dblocate) では、代替サーバー名も検出されます。

### 参照


- 「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』
- 「START DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「データベースミラーリング」1137 ページ
- 「サーバー列挙ユーティリティ (dblocate)」999 ページ
- AlternateServerName データベースプロパティ782 ページ


### 例

次のコマンドは、データベースサーバー myserver 上のデータベース *satest.db* と *sample.db* を起動します。-sn オプションを指定すると、データベースサーバーは *sample.db* に接続するときに代替サーバー名として mysample を使用します。

```
dbsrv12 -n myserver satest.db sample.db -sn mysample
```

*sample.db* に接続するには、次のいずれかの接続パラメーターを使用します。

- Server=myserver;DBN=sample
- Server=mysample
- Server=mysample;DBN=sample

Server=mysample を使用して *satest.db* に接続することはできません。

## -wc dbeng12/dbsrv12 データベースオプション

チェックサムがデフォルトで有効になっていない場合に、データベースの書き込み操作でチェックサムが有効になっているかどうかを制御します。

### 構文

{ **dbeng12** | **dbsrv12** } [ *server-options* ] *database-file* **-wc**[ **+** | **-** ]

### 適用対象

すべてのオペレーティングシステムとサーバー

### 備考

書き込みチェックサム (-wc オプションで有効) とグローバルチェックサム (チェックサムが有効な状態でデータベースを作成) の違いは、-wc ではデータベースページがディスクに書き込まれるときにのみチェックサムが行われる点です。ディスクから読み込まれるページは、ページが書き込まれる前にチェックサム値が計算された場合にのみ検証されます。データベースでチェックサムが有効になっている場合、ページが書き込まれるときにすべてのページに対してチェックサムが計算され、ページが読み込まれるときにすべてのページに対してチェックサムが検証されます。

データベースが Windows Mobile またはリムーバブルストレージデバイス (ネットワーク共有や USB デバイスなど) で実行されていることがデータベースサーバーで検出されると、データベースサーバーでは、すべてのデータベースページに対してグローバルチェックサムが自動的に有効になります。

デフォルトでは、バージョン 10 または 11 の SQL Anywhere で作成されたデータベースではグローバルチェックサムは有効になっていません。バージョン 12 以降のデータベースサーバーで SQL Anywhere 11 を使用して作成されたデータベースを起動すると、デフォルトでは、ページがディスクに書き込まれるときにデータベースサーバーでページに対して書き込みチェックサムが作成されます (-wc+)。バージョン 12 以降のデータベースではデフォルトでグローバルチェックサムが有効になっています。デフォルトではすべてのデータベースページにチェックサムがあるため、これらのデータベースに対してデータベースサーバーのデフォルト値は -wc- になります。デフォルトのチェックサム設定を使用しない場合は、-wc オプションまたは START DATABASE 文を使用してデータベースサーバーのチェックサム動作を変更できます。

次の文を実行することによって、データベースがグローバルチェックサムを有効にして作成されたかどうかをチェックできます。

```
SELECT DB_PROPERTY ( 'Checksum' );
```

次の文を実行することによってのみ、書き込み I/O 操作に対して書き込みチェックサムが有効になっているかどうかをチェックできます。

```
SELECT DB_PROPERTY ( 'WriteChecksum' );
```

### 参照


- 「チェックサムを使用した破損の検出」1120 ページ
- 「-wc dbeng12/dbsrv12 サーバーオプション」288 ページ
- 「START DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』


## -xp dbsrv12 データベースオプション

データベースミラーリングまたは読み込み専用のスケールアウトが使用されている場合に、パートナーサーバーと監視サーバーへの接続を可能にする情報をサーバーに提供します。**-xp on** 以外のすべての構文は使用されなくなりました。

### 構文

**dbsrv12** [ *server-options* ] *database-file* **-xp on**

### 構文 (旧式)

**dbsrv12** [ *server-options* ] *database-file*
**-xp** { **on** | *mirror-settings* }

*mirror-settings* :
**partner=**( *partner-conn* );
**auth=***auth-str*;
[ ;**arbiter=**( *arbiter-conn* ) ]
[ ;**mode=**[ **sync** | **async** | **page** ]
[ ;**autofailover=**[ **YES** | **NO** ] ]
[ ;**pagetimeout=***n* ]
[ ;**preferred=**[ **YES** | **NO** ] ...

### 適用対象

すべてのオペレーティングシステム (Windows Mobile を除く)、ネットワークサーバーのみ

### 備考

-xp を指定する場合、-xf オプションも指定して、データベースミラーリングのステータス情報ファイルのロケーションを指定する必要があります。-xp データベースオプションは、*database-file* の後に指定する必要があり、そのデータベースだけに適用されます。

-xp オプションで指定した接続パラメーターが無効であり、サーバーで複数のデータベースが実行されている場合、ミラーデータベースの起動に失敗し、再接続は試行されません。ミラーデータベースが、データベースサーバーで実行されている唯一のデータベースである場合は、データベースサーバーが起動しません。

**on**　ミラーリング情報またはスケールアウト情報がデータベースに格納されていても、データベースサーバーの起動時に -xp オプションを指定した場合にのみデータベースミラーリングやスケールアウトを使用できます。-xp on を指定すると、ミラーリングまたはスケールアウトの他

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

のオプションは -xp オプションで指定できません。値 off はサポートされていません。データベースミラーリングとスケールアウトの設定は、次の文を使用してデータベースに定義されます。

○ CREATE MIRROR SERVER
○ SET MIRROR OPTION

**-xp on** を指定するときには、-n オプションを使用してミラーリングシステムでデータベースサーバーの名前も指定します。また、**-su** オプションも使用して、ユーティリティデータベースのパスワードを指定することをおすすめします。これにより、ユーティリティデータベースを使用してデータベースサーバーをシャットダウンしたり、必要に応じてミラーサーバーを強制的にプライマリサーバーにしたりできるようになります。

**partner-conn (旧式)**　パートナーサーバーの接続文字列を指定します。ユーザー ID とパスワードは必要ありません。フェールオーバー時間を短縮するためにタイムアウトを指定するようおすすめします。

**auth-str (旧式)**　監視サーバーが使用する認証文字列を指定します。

**arbiter-conn (旧式)**　監視サーバーの接続文字列を指定します。ユーザー ID とパスワードは必要ありません。フェールオーバー時間を短縮するためにタイムアウトを指定するようおすすめします。

**mode (旧式)**　データベースミラーリングで使用する同期実行モードとして、同期 (sync)、非同期 (async)、非同期フルページ (page) のいずれかを指定します。

**autofailover (旧式)**　プライマリサーバーで障害が発生した場合にミラーサーバーが自動的にプライマリサーバーになるかどうかを指定します。このオプションは同期モードには適用されません。

**注意**
非同期モードまたは非同期フルページモードを使用している場合は、-xp autofailover オプションを Yes に設定することをおすすめします。それによって、プライマリサーバーで障害が発生した場合、ミラーサーバーが自動的にプライマリサーバーとなります。

**pagetimeout (旧式)**　トランザクションログページを満杯かどうかにかかわらずミラーサーバーに送信する間隔を秒単位で指定します。このオプションは、非同期フルページモードを使用している場合のみ適用されます。

**preferred (旧式)**　サーバーがミラーリングシステムにおいて優先サーバーであるかどうかを指定します。優先サーバーは、可能なかぎり、プライマリサーバーのロールを引き受けます。これは、CREATE MIRROR SERVER 文で優先オプションを指定することと同じです。「優先データベースサーバー」1164 ページを参照してください。

**警告**
非同期モードと非同期フルページモードでは、コミットされたトランザクションがミラーサーバーに記録されることは保証されず、データが失われる可能性があります。「データベースミラーリングモード」1142 ページを参照してください。

**参照**

- 「CREATE MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』
- 「SET MIRROR OPTION 文」『SQL Anywhere サーバー SQL リファレンス』
- 「データベースミラーリングシステムの設定」1160 ページ
- 「SQL Anywhere の読み込み専用のスケールアウト」1188 ページ
- 「チュートリアル：読み込み専用のスケールアウトシステムの作成とモニタリング」1190 ページ
- 「チュートリアル：データベースミラーリングの使用」1146 ページ
- 「チュートリアル:監視サーバーを共有する複数のデータベースでデータベースミラーリングを使用する」1154 ページ
- 「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』
- 「データベースミラーリングモード」1142 ページ
- 「-sn dbsrv12 オプション」315 ページ
- 「-su dbeng12/dbsrv12 サーバーオプション」276 ページ
- 「-xa dbsrv12 サーバーオプション」290 ページ
- 「-xf dbsrv12 サーバーオプション」292 ページ
- 「-n dbeng12/dbsrv12 サーバーオプション」257 ページ
- MirrorMode データベースプロパティ797 ページ

**例**

次のコマンドは、ミラーリングシステムに参加できるデータベースサーバーで 3 つのデータベースファイルを起動します。

```
dbsrv12 -n server1 -x tcpip(PORT=6871) -su sql
c:¥server1¥one.db -xp on c:¥server1¥two.db -xp on c:¥server1¥three.db -xp on
```

# 接続パラメーター

接続パラメーターは、接続文字列に含めます。次の場所で入力できます。

- アプリケーションの接続文字列。「接続パラメーターリストのアセンブル」78 ページと「接続文字列として渡される接続パラメーター」31 ページを参照してください。

- SQLConnect 環境変数

- ODBC データソース。「ODBC データソース」36 ページを参照してください。

- SQL Anywhere の **[接続]** ウィンドウ。「SQL Anywhere ユーティリティの接続」29 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**注意**

- 値に大文字と小文字が区別されるもの (UNIX のファイル名など) が含まれている場合でも、接続パラメーターは大文字と小文字を区別しません。
- ブールパラメーターは、YES、Y、ON、TRUE、T、1 のいずれかによってオンになり、NO、N、OFF、FALSE、F、0 のいずれかによってオフになります。パラメーターは、大文字と小文字を区別しません。
- 各接続パラメーターの使用法のところで、パラメーターが使用される状況を説明します。一般的に使用法にあげる項目は以下のとおりです。
  - **組み込みデータベース**　SQL Anywhere を組み込みデータベースとして使用した場合、接続するとパーソナルサーバーが起動し、データベースがロードされます。アプリケーションがデータベースから切断されると、データベースはアンロードされ、サーバーが停止します。
  - **実行中のローカルデータベース**　これは、SQL Anywhere パーソナルサーバーがすでに実行中で、データベースがすでにサーバーにロードされている場合を指します。
  - **ネットワークサーバー**　SQL Anywhere をネットワークサーバーとして使用する場合、クライアントアプリケーションはネットワーク上ですでに実行しているサーバーを検出し、データベースに接続します。
- dbping ユーティリティを使用して、接続文字列をテストできます。-c オプションを使用して、接続パラメーターを指定します。たとえば、demo12 という名前のパーソナルサーバーがサンプルデータベース (コマンド dbeng12 "%SQLANYSAMP12%¥demo.db" で実行可能) を実行しているとします。次のコマンドは、データベースへの接続が成功したというメッセージを返します。

```
dbping -d -c "Server=demo12;DBN=demo;UID=DBA;PWD=sql"
```

ローカルコンピューター上で other-server という名前のデータベースサーバーが実行中ではない場合、次のコマンドは、「**データベースへの ping が失敗しました - データベースサーバーは起動していません。**」というメッセージを返します。

```
dbping -c "Server=other-server"
```

**参照**


- 「Ping ユーティリティ (dbping)」994 ページ
- 「接続パラメーターと接続文字列」30 ページ
- 「接続パラメーターの構文ルール」34 ページ


# AppInfo (APP) 接続パラメーター

データベースサーバーからの特定のクライアント接続の開始を、管理者が容易に識別できるようにします。

### 構文

{ **AppInfo** | **APP** }=*keyword*=*value*

### 使用法

特に制限なし

### 指定可能な値

クライアントは、固有の文字列を指定できます。指定した文字列は、生成された文字列に追加されます。AppInfo プロパティ文字列は、セミコロンで区切られた **key**=*value* ペアのシーケンスです。有効なキーは次のとおりです。

- **API**　DBLIB、ODBC、OLEDB、ADO.NET、iAnywhereJDBC、PHP、PerlDBD、DBEXPRESS のいずれか
- **APPINFO**　接続文字列に AppInfo を指定したときに入力される文字列
- **EXE**　クライアント実行プログラムの名前 (Windows、Linux、Solaris)
- **HOST**　クライアントコンピューターのホスト名
- **IP**　クライアントコンピューターの IP アドレス
- **OS**　オペレーティングシステム名とバージョン番号 (Windows XP など)
- **OSUSER**　クライアントプロセスに関連付けられたオペレーティングシステムユーザー名。クライアントプロセスで別のユーザーを同一化している場合 (UNIX の場合は、セット ID ビットが設定されている場合)、同一化されたユーザー名が返されます。バージョン 10.0.1 以前のクライアントと HTTP や TDS クライアントでは、空の文字列を返します。
- **PID**　クライアントのプロセス ID (Windows と UNIX のみ)
- **THREAD**　クライアントのスレッド ID (Windows と UNIX のみ)
- **TIMEZONEADJUSTMENT**　接続のローカル時間を表示するために協定世界時 (UTC: Coordinated Universal Time) に加算する必要がある分数
- **VERSION**　主要なバージョン番号、それ以外のバージョン番号、ビルド番号を含む、使用しているクライアントライブラリのバージョン (12.0.1.3600 など)

### デフォルト

空の文字列

### 備考

この接続パラメーターは、Embedded SQL、ODBC、OLE DB、または ADO.NET クライアント、および SQL Anywhere JDBC ドライバーを使用するアプリケーションから、データベースサーバーに送信されます。Sybase Open Client または jConnect アプリケーションから使用することはできません。

このパラメーターは、クライアントコンピューターの IP アドレスや実行されているオペレーティングシステムなどの、クライアントプロセスについての情報を保持するように生成された文

字列で構成されています。文字列は、接続するデータベースサーバーに関連付けられており、次の文を使用して検索できます。

```
SELECT CONNECTION_PROPERTY( 'AppInfo' );
```

クライアント接続パラメーターにデバッグログファイルを指定すると、そのファイルに AppInfo 文字列が追加されます。

### 参照


- 「接続パラメーターと接続文字列」30 ページ
- 「request_timeout オプション」672 ページ
- AppInfo 接続プロパティ715 ページ


### 例

Interactive SQL からのサンプルデータベースに接続します (デフォルトで SQL Anywhere JDBC ドライバーを使用します)。

```
dbisql -c "UID=DBA;PWD=sql;DBF=%SQLANYSAMP12%¥demo.db"
```

アプリケーション情報を表示します。

```
SELECT CONNECTION_PROPERTY( 'AppInfo' );
```

結果は、次のとおりです (1 つの文字列で表示されます)。

```
IP=ip-address;
HOST=computer-name;
OSUSER=user-name;
OS='Windows XP Build 2600 Service Pack 2';
EXE='C:¥Program Files¥SQL Anywhere 12¥Bin32¥dbisql.exe';P
ID=0xcac;
THREAD=0xca8;VERSION=12.0.1.3600;
API=iAnywhereJDBC;
TIMEZONEADJUSTMENT=-240
```

AppInfo プロパティにユーザー固有の情報を追加して、Interactive SQL からサンプルデータベースに接続します。

```
dbisql -c "UID=DBA;PWD=sql;DBF=%SQLANYSAMP12%¥demo.db;APP=Interactive SQL connection"
```

アプリケーション情報を表示します。

```
SELECT CONNECTION_PROPERTY( 'AppInfo' );
```

結果は、次のとおりです (1 つの文字列で表示されます)。

```
IP=ip-address;
HOST=computer-name;
OSUSER=user-name;
OS=Windows XP Build 2600 Service Pack 2;
EXE=C:¥Program Files¥SQL Anywhere 12¥Bin32¥dbisql.exe;
PID=0xcac;
THREAD=0xba8;
VERSION=12.0.1.3600;
API=iAnywhereJDBC;
TIMEZONEADJUSTMENT=-240;
APPINFO='Interactive SQL connection'
```

# AutoStart (ASTART) 接続パラメーター

接続可能なサーバがない場合にローカルデータベースサーバーを起動するかどうかを制御します。

### 構文

**{ AutoStart | ASTART }={ YES | NO }**

### 使用法

特に制限なし

### デフォルト

YES

### 備考

データベースファイル、データベース名、START 接続パラメーターのいずれかが指定されていて、接続時にデータベースサーバーが検出できない場合、デフォルトでは、パーソナルデータベースサーバーは同一のコンピューターで起動されます。接続文字列の AutoStart (ASTART) 接続パラメーターを NO に設定することによって、このような動作を無効にできます。Host 接続パラメーターが指定されているか、CommLinks (LINKS) パラメーターに TCPIP が含まれている場合、データベースサーバーは自動的には起動しません。

自動的に起動されるデータベースのクエリのパフォーマンスを向上するには、ユーザーがすぐに接続しない場合でも、データベースをできるだけ早く起動します。このように起動することで、データベースに対してクエリが実行される前にキャッシュの準備が完了します。「キャッシュウォーミング」『SQL Anywhere サーバー SQL の使用法』を参照してください。

> **注意**
> AutoStart 接続パラメータは、サーバーを自動的に起動させるかどうかを制御します。ただし、自動的に起動させない場合は、データベースを自動的に起動させるかどうかを決定します。データベースが自動的に起動しないようにするには、DatabaseFile 接続パラメーターではなく DatabaseName 接続パラメーターを使用します。「DatabaseName (DBN) 接続パラメーター」336 ページを参照してください。

### 参照


- 「接続パラメーターと接続文字列」30 ページ
- 「データベースサーバーの検出」80 ページ
- 「CommLinks (LINKS) 接続パラメーター」327 ページ
- 「Elevate 接続パラメーター」340 ページ


# AutoStop (ASTOP) 接続パラメーター

オープンな非 HTTP 接続がなくなったときにデータベースを停止するかどうかを制御します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 構文

**{ AutoStop | ASTOP }={ YES | NO }**

### 使用法

組み込みデータベース

### デフォルト

YES

### 備考

デフォルトでは、接続文字列で起動したすべてのデータベースサーバーは、非 HTTP 接続がなくなると停止します。また、接続文字列からロードしたすべてのデータベースも、非 HTTP 接続がなくなるとアンロードされます。この動作は、AutoStop=YES と同じです。

AutoStop=NO を指定すると、その接続で起動したすべてのデータベースは、非 HTTP 接続がなくなっても実行を続けます。したがって、データベースサーバーは操作可能な状態を維持します。

データベースへの唯一の接続が HTTP 接続で、自動的に停止するようにデータベースが設定されている場合、HTTP 接続が切断したときにデータベースは自動的に停止しません。また、自動的に停止するように設定されているデータベースに HTTP 接続と Command Sequence 接続または TDS 接続がある場合は、最後の Command Sequence 接続または TDS 接続が切断したときにデータベースが停止します。このときに HTTP 接続がまだあった場合は切断されます。 「-ga dbeng12/dbsrv12 サーバーオプション」231 ページと「AutoStop (ASTOP) 接続パラメーター」324 ページを参照してください。

AutoStop (ASTOP) 接続パラメーターは、現在実行していないデータベースに接続するときにのみ使用されます。データベースがすでに起動されている場合は、無視されます。

.NET アプリケーションで AutoStop 接続パラメーターを使用する場合は注意が必要です。接続を閉じると関連するアプリケーションも閉じられますが、接続プールが有効な場合、アクティブな接続は開いたままになります。その結果、サーバーの停止を予期していても、そのように動作しません。

### 参照


- 「接続パラメーターと接続文字列」30 ページ
- 「接続プーリング」『SQL Anywhere サーバー プログラミング』
- 「データベースの開始と停止」147 ページ
- 「START DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』


## CharSet (CS) 接続パラメーター

この接続で使用する文字セットを指定します。

### 構文

**{ CharSet | CS }={ NONE |** *character-set* **}**

### 使用法

特に制限なし

### 指定可能な値

- **NONE**　CharSet=NONE を指定すると、接続でデータベース CHAR 文字セットの使用することを要求します。
- **character-set**　使用する文字セットラベルを指定します。 有効な文字セット値の詳細については、「推奨文字セットと照合」506 ページを参照してください。

### デフォルト

ローカル文字セット

ローカル文字セットがどのように決定されるかについては、「ロケール情報」501 ページを参照してください。

### 備考

CharSet に値を指定すると、指定された文字セットが現在の接続に使用されます。

データをアンロードする場合は、CharSet 接続パラメーターを使用して文字セットを指定します。

Unicode クライアント API を使用している場合は、文字セット変換によるデータの損失を避けるために、CharSet 接続パラメーターを設定することはおすすめできません。Unicode クライアント API には、ADO.NET、OLE DB、SQL Anywhere JDBC ドライバーが含まれます。ODBC も、ワイド (Unicode) 関数が使用されている場合は、Unicode クライアント API です。

### 参照


- 「接続パラメーターと接続文字列」30 ページ
- 「SACHARSET 環境変数」449 ページ
- 「ロケール文字セット」487 ページ


### 例

次の接続文字列フラグメントは、Windows コードページ 1252 を接続に使用することを指定します。

```
CharSet=windows-1252
```

## CommBufferSize (CBSIZE) 接続パラメーター

通信パケットの最大サイズを設定します。

### 構文

**{ CommBufferSize | CBSIZE }=***size*[ **k** ]

### 使用法

特に制限なし


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 指定可能な値

- **size**　この整数は、通信パケットの最大サイズを指定します。デフォルト値はバイト単位ですが、**k** を使用してキロバイトの単位を指定できます。CommBufferSize の最小値は 500 バイトで、最大値は 16000 バイトです。

### デフォルト

CommBufferSize 値が設定されていない場合、CommBufferSize は、サーバー側の設定によって制御されます。Windows Mobile 以外のすべてのオペレーティングシステムでのデフォルト値は 7300 バイトです。Windows Mobile の場合、デフォルト値は 1460 バイトです。

### 備考

ネットワーク上のパケットの最大サイズは、プロトコルスタックによって設定されます。CommBufferSize をネットワークで許可されているサイズより大きく設定すると、通信パケットがネットワークソフトウェアによって分割されます。デフォルトのサイズは、標準 Ethernet TCP/IP の最大パケットサイズ (1460 バイト) の倍数です。

パケットサイズを大きくすると、複数のローのフェッチと長いローのフェッチのパフォーマンスが向上しますが、クライアントとサーバーのメモリ使用量が増加します。

クライアント側で CommBufferSize の指定がないと、接続ではサーバーのバッファーサイズが使用されます。クライアント側で CommBufferSize の指定がある場合、接続では CommBufferSize 値が使用されます。

-p データベースサーバーオプションを使用して CommBufferSize を設定すると、CommBufferSize を指定していないすべてのクライアントで -p データベースサーバーオプションで指定されたサイズが使用されます。

### 参照


- 「接続パラメーターと接続文字列」30 ページ
- 「TCP/IP パフォーマンス」95 ページ
- 「-p dbeng12/dbsrv12 サーバーオプション」262 ページ


### 例

バッファーサイズを 1460 バイトに設定します。

```
...
CommBufferSize=1460
...
```

またこのパラメーターは、**[SQL Anywhere の ODBC 設定]** ウィンドウの **[詳細]** タブで、**[CommBufferSize]** テキストボックスに値を入力して設定することもできます。

## CommLinks (LINKS) 接続パラメーター

クライアント側のネットワークプロトコルオプションを指定します。

### 構文

{ **CommLinks** | **LINKS** }={ [ **SharedMemory** | **ShMem** ] | **ALL** | [ **TCPIP** | **TCP** ] } [, ... ] *string*

### 使用法

CommLinks (LINKS) 接続パラメーターは、ネットワークデータベースサーバーへの接続に使用します。オプションで、パーソナルデータベースサーバーへの接続にも使用します。

**注意**
CommLinks (LINKS) 接続パラメーターは、HOST または ServerPort (PORT) 以外の TCP/IP プロトコルオプションを指定する必要がある場合のみ使用することをおすすめします。これ以外の場合は、Host 接続パラメーターを使用します。「Host 接続パラメーター」346 ページを参照してください。

接続文字列に CommLinks と Host の両方は指定できません。

バージョン 10.0.0 から 12.0.0 のクライアント ライブラリを使用してクラウドのデータベースに接続する場合は、これらのクライアントライブラリでは接続のリダイレクトがサポートされていないため、CommLinks (LINKS) 接続パラメーターを使用する必要があります。

### 指定可能な値

CommLinks (LINKS) 接続パラメーターの値は、大文字と小文字を区別しません。次の値が含まれます。

- **SharedMemory (ShMem)**　同一コンピューター通信の共有メモリプロトコルを使用して接続します。これがデフォルト設定です。共有メモリが CommLinks(LINKS) 接続パラメーターによって指定されたプロトコルのリストに含まれている場合は、リストでの順序に関係なく、クライアントは最初に共有メモリを使用しようとします。

- **ALL**　最初に共有メモリプロトコルを使用して接続しようとし、次に TCP/IP を使用します。

- **TCPIP (TCP)**　TCP/IP 通信プロトコルを使用して接続します。TCP/IP は、すべてのオペレーティングシステムでサポートされています。

  TCPIP 用の追加のネットワークプロトコルオプションを指定できます。「ネットワークプロトコルオプション」369 ページを参照してください。

### デフォルト

SharedMemory

### 備考

Host 接続パラメーターまたは CommLinks (LINKS) 接続パラメーターを指定しないと、クライアントは現在のコンピューターにあるデータベースサーバーしか検索せず、共有メモリ接続のみを使用します。これはデフォルトの動作であり、CommLinks=ShMem を指定するのと同じです。共有メモリプロトコルは、パーソナルデータベースサーバーに接続するアプリケーションでの標準的な使い方で、同じコンピューターで実行されているクライアントとデータベースサーバー間で最速の通信リンクです。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

UNIX での共有メモリ接続の保護については、「セキュリティに関するヒント」1380 ページを参照してください。

TCPIP と SharedMemory の両方、または CommLinks=ALL を指定すると、最初に共有メモリが使用され、共有メモリでデータベースサーバーが検出できない場合に TCP/IP が使用されます。たとえば、次の接続文字列では、共有メモリが先に使用されます。

```
"UID=DBA;PWD=sql;Server=demo;DBN=demo;LINKS=TCPIP,SHMEM"
```

Host 接続パラメーターが指定されているか、CommLinks (LINKS) パラメーターに TCP/IP が含まれている場合、パーソナルデータベースサーバーは自動的には起動しません。

-x オプションを使用すると、データベースサーバーで TCP/IP 通信プロトコルが有効になります。

### 参照


- 「Host 接続パラメーター」346 ページ
- 「ネットワークプロトコルオプション」369 ページ
- 「通信プロトコル」96 ページ
- 「-x dbeng12/dbsrv12 サーバーオプション」289 ページ
- 「接続パラメーターと接続文字列」30 ページ
- 「高速の接続のための sasrv.ini でのデータベースサーバーアドレス情報のキャッシング」91 ページ
- CommLink 接続プロパティ719 ページ


### 例

次の接続文字列フラグメントは、現在のサブネット上で実行中の demo という名前のデータベースサーバーに接続しようとしています。

```
CommLinks=tcpip;Server=demo;UID=DBA;PWD=sql;
```

次の接続文字列フラグメントは、demo という名前のデータベースサーバーに接続しようとしています。最初は、共有メモリを介して接続しようとします。共有メモリを使用してもデータベースサーバーが見つからない場合、TCP/IP を使用して接続しようとします。

```
CommLinks=tcpip,shmem;Server=demo;UID=DBA;PWD=sql
```

次の接続文字列は、kangaroo という名前のホスト上で実行している demo という名前のサーバーに接続しようとします。

```
CommLinks=tcpip(HOST=kangaroo);Server=demo;UID=DBA;PWD=sql
```

kangaroo という名前のホスト上で実行している demo という名前のサーバーへは、Host 接続パラメーターを使用して接続することも可能です。

```
Host=kangaroo;Server=demo;UID=DBA;PWD=sql
```

## Compress (COMP) 接続パラメーター

接続の圧縮をオンまたはオフに設定します。

### 構文

**{ Compress | COMP }={ YES | NO }**

### 使用法

TDS 接続以外。他には特に制限なし。TDS 接続 (jConnect を含む) では、SQL Anywhere 通信圧縮はサポートされません。

### デフォルト

NO

### 備考

SQL Anywhere クライアントとデータベースサーバーの間でやり取りされるパケットは、Compress (COMP) 接続パラメーターを使用して圧縮できます。状況によっては、接続を圧縮することでパフォーマンスが向上する場合があります。大幅に圧縮可能なデータの大規模なデータ転送では、圧縮率が高くなります。

Compress 接続パラメーターに値が設定されていない場合、圧縮ステータスはデータベースサーバー側の設定によって制御されます。デフォルトでは、圧縮を行いません。クライアントとデータベースサーバーで設定が異なる場合は、クライアントの設定が適用されます。

特定のアプリケーションを使用してネットワークのパフォーマンス分析を行ってから、運用環境で通信の圧縮を使用することをおすすめします。

データベースサーバーのすべてのリモート接続で圧縮を有効にするには、-pc サーバーオプションを使用します。

-pc オプションまたは COMPRESS=YES パラメーターを指定しても、使用する通信リンクにかかわらず、同一コンピューター接続では圧縮は有効になりません。

### 参照


- 「接続パラメーターと接続文字列」30 ページ
- 「-pc dbsrv12 サーバーオプション」263 ページ
- 「通信の圧縮設定」106 ページ
- 「圧縮の慎重な使用」『SQL Anywhere サーバー SQL の使用法』


### 例

次の接続文字列フラグメントは、パケット圧縮をオンに設定します。

```
Compress=YES
```

次の接続文字列フラグメントは、パケット圧縮をオフに設定します。

```
Compress=NO
```

## CompressionThreshold (COMPTH) 接続パラメーター

パケットの圧縮が適用される最小パケットサイズを増減します。

### 構文

{ **CommBufferSize** | **COMPTH** }=*size*[ **k** ]

### 使用法

TDS 以外。他には特に制限なし。圧縮接続にのみ適用。

### 指定可能な値

- **size**　この整数は、パケットの圧縮が適用される最小パケットサイズを指定します。デフォルト値はバイト単位ですが、**k** を使用してキロバイトの単位を指定できます。CommBufferSize の最小値は 500 バイトで、最大値は 16000 バイトです。クライアントとデータベースサーバーで圧縮スレッショルドの設定が異なる場合は、クライアントの設定が適用されます。サポートされている最小値は 1 バイト、最大値は 32767 バイトです。80 バイト未満の値はおすすめしません。

### デフォルト

120

CompressionThreshold 値が設定されていない場合、圧縮スレッショルド値は、サーバー側の設定によって制御されます。デフォルトは 120 バイトです。

### 備考

圧縮した場合に転送速度が速くなるパケットのみを圧縮するように圧縮スレッショルドを変更して、圧縮接続のパフォーマンスを向上できます。

圧縮が有効な場合、パケットは、各々のサイズに応じて圧縮するかどうかを決定します。たとえば、SQL Anywhere では、圧縮のスレッショルドよりも小さいパケットは、通信の圧縮が有効な場合でも圧縮されません。同様に、小さなパケット (100 バイト未満) は、通常はまったく圧縮されません。パケットの圧縮には CPU 時間が必要なので、小さなパケットを圧縮しようとすると、実際にパフォーマンスが低下することがあります。

圧縮スレッショルドの値を小さくすると、非常に低速なネットワークではパフォーマンスが向上し、値を大きくすると CPU 使用率の減少によってパフォーマンスが向上する場合があります。ただし、圧縮のスレッショルド値を小さくするとクライアントとサーバーの両方で CPU 使用率が増加するので、パフォーマンス分析を行って、圧縮のスレッショルドを変更することでパフォーマンスが向上するかどうかを判断してください。

### 参照


- 「-pt dbsrv12 サーバーオプション」264 ページ
- 「通信の圧縮設定」106 ページ
- 「接続パラメーターと接続文字列」30 ページ


### 例

圧縮スレッショルド値を 100 バイトに設定して接続します。

```
CompressionThreshold=100
```

# ConnectionName (CON) 接続パラメーター

接続に名前を付け、マルチ接続アプリケーションで簡単に切り替えができるようにします。

### 構文

{ **ConnectionName** | **CON** }=*connection-name*

### 使用法

特に制限なし

### 指定可能な値

- **connection-name**　この文字列は、確立中の特定の接続に名前を指定します。

### デフォルト

接続名なし

### 備考

この接続パラメーターはオプションです。複数の接続を確立しても切り替えを行わないときは、この値を指定する必要はありません。

接続名はデータソース名とは異なります。

### 参照


- 「接続パラメーターと接続文字列」30 ページ
- 「SET CONNECTION statement [Interactive SQL] [ESQL]」『SQL Anywhere サーバー SQL リファレンス』
- Name 接続プロパティ 732 ページ


### 例

first-con という名前の接続を指定して、データベースに接続します。

```
CON=first-con
```

# ConnectionPool (CPOOL) 接続パラメーター

クライアントの接続プールの動作を制御します。

### 構文

**ConnectionPool=**{ **NO** | **YES** [ ( [ **Timeout=***timeout-sec*; ] [ **MaxCached=***max-cached-conn* ] ) ] }

### 使用方法

Windows Mobile と非スレッド型 UNIX クライアントを除くすべてのプラットフォーム


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 指定可能な値

- **timeout-sec**　接続プールのアイドルタイムアウト時間 (秒単位)。デフォルト値は 60 秒です。キャッシュされた接続は、*timeout-sec* で指定された時間内に再利用しないと、再利用できなくなります。

- **max-cached-conn**　各アプリケーションのキャッシュされた接続の最大数。デフォルトの接続数は 5 です。接続が切断され、*max-cached-conn* で指定された最大接続数に達していない場合は、接続がキャッシュされます。接続が再初期化されます。接続はアプリケーションによって切断されますが、キャッシュされた接続はデータベースサーバーに接続したままになります。実行されるクリーンアップと再初期化のタスクの詳細については、「接続プーリング」74 ページを参照してください。

### デフォルト

YES

### 備考

接続プールによって、データベースサーバーに対して複数の短時間接続を行うアプリケーションのパフォーマンスが向上する可能性があります。接続が切断されると自動的にキャッシュされ、アプリケーションが再接続した時点で再利用できます。接続をプールするには、接続名は異なっていてもかまいませんが、他のすべての接続パラメーターは同じである必要があります。

### 参照

- 「接続プーリング」74 ページ
- 「接続パラメーターと接続文字列」30 ページ
- ConnPoolCachedCount データベースプロパティ 787 ページ
- ConnPoolHits データベースプロパティ 787 ページ
- ConnPoolMisses データベースプロパティ 787 ページ

### 例

次の接続文字列フラグメントは、接続プールをオフに設定します。

```
ConnectionPool=NO;
```

次の接続文字列フラグメントは、キャッシュされた接続の最大数に 10 を使用して、接続プールをオンに設定します。

```
CPOOL=YES(MaxCached=10);
```

## DatabaseFile (DBF) 接続パラメーター

実行していないデータベースを起動したときにロードして接続するデータベースファイルを指定します。

### 構文

{ **DatabaseFile** | **DBF** }=*filename*

### 使用法

組み込みデータベース

### 指定可能な値

- **filename**　この文字列は、すでに実行しているデータベースのパスとファイル名を指定します。

  ○ファイル名に拡張子が含まれていない場合、SQL Anywhere は *.db* 拡張子の付いたファイルを検索します。

  ○ファイルのパスは、データベースサーバーの作業ディレクトリの相対パスです。データベースサーバーをコマンドプロンプトで起動すると、コマンド入力時の (現在の) ディレクトリが作業ディレクトリになります。データベースサーバーをアイコンかショートカットから起動すると、アイコンかショートカットを指定したディレクトリが作業ディレクトリになります。完全なパスとファイル名を指定することをおすすめします。

  ○データベースファイルとデータベース名の両方を指定すると、指定した名前の実行中のデータベースに接続します (データベースファイルは無視されます)。接続に失敗すると、データベースファイルとデータベース名の両方を使用して、データベースが自動的に起動されます。データベースサーバーは自動的に起動されません。

  ○UNC ファイル名がサポートされています。

  UNC ファイル名の詳細については、「SQL Anywhere データベースサーバーの構文」189 ページを参照してください。

### デフォルト

デフォルト設定はありません。

### 備考

接続するデータベースが実行されていない場合、DatabaseFile (DBF) 接続パラメーターを使用してそのデータベースを起動できます。

実行されていないデータベースファイルを自動的に起動する場合、配備されたアプリケーションでは、ServerName (Server) パラメーターを使用してデータベースサーバー名を指定することをおすすめします。そうしないと、アプリケーションが別のデータベースサーバーに接続する可能性があります。たとえば、データベースサーバーは、すでに実行されている埋め込みアプリケーションの一部である、異なるバージョンの SQL Anywhere サーバーに接続することがあります。

**警告**
データベースファイルは、データベースサーバーと同じコンピューター上に置いてください。ネットワークドライブにあるデータベースファイルを起動すると、ファイルが破損することがあります。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照


- 「DatabaseFile (DBF) 接続パラメーターの使用」28 ページ
- 「-gd dbeng12/dbsrv12 サーバーオプション」233 ページ
- 「CommLinks (LINKS) 接続パラメーター」327 ページ
- 「DatabaseName (DBN) 接続パラメーター」336 ページ
- 「接続パラメーターと接続文字列」30 ページ
- 「組み込みデータベースの接続」28 ページ


### 例

次の例の DatabaseFile (DBF) 接続パラメーターは、サンプルデータベース *demo.db* をロードして接続します。

```
"DBF=%SQLANYSAMP12%¥demo.db"
```

次の 2 つの例では、*cities.db* というデータベースファイルをすでに起動していて、次のように名前を Kitchener に変更したと仮定します。

```
dbeng12 cities.db -n Kitchener
```

データベースを起動して接続し、それを Kitchener と命名するには、次の手順に従います。

```
DBN=Kitchener;DBF=cities.db
```

DBF=cities.db と指定すると、Kitchener という名前の実行中のデータベースへの接続が失敗します。

## DatabaseKey (DBKEY) 接続パラメーター

暗号化されたデータベースを接続要求で起動します。

### 構文

{ **DatabaseKey** | **DBKEY** }=*key*

### 使用法

特に制限なし。

### 指定可能な値

- **key**　暗号化キーは、大文字、小文字、数字、文字、特殊記号を含む文字列です。データベースキーには、前後のスペースやセミコロンを含めることはできません。

### デフォルト

なし

### 備考

接続要求で暗号化データベースを起動するときには、このパラメーターを指定します。このパラメーターを指定する必要はありません。

クライアントアプリケーションとデータベースサーバー間の通信パケットを安全化するには、-ec サーバーオプションとトランスポートレイヤーセキュリティを使用します。

### 参照


- 「トランスポートレイヤーセキュリティ」1410 ページ
- 「トランスポートレイヤーセキュリティを使用するようにクライアントアプリケーションを設定する」1422 ページ
- 「-ec dbeng12/dbsrv12 サーバーオプション」223 ページ
- 「-ek dbeng12/dbsrv12 データベースオプション」310 ページ
- 「-ep dbeng12/dbsrv12 サーバーオプション」226 ページ
- 「-es dbeng12/dbsrv12 サーバーオプション」227 ページ
- 「接続パラメーターと接続文字列」30 ページ
- 「Encryption (ENC) 接続パラメーター」342 ページ


### 例

次のフラグメントは、DatabaseKey (DBKEY) 接続パラメーターの使用方法を示します。

```
"UID=DBA;PWD=sql;Server=myeng;DBKEY=V3moj3952B;DBF=%SQLANYSAMP12%¥demo.db"
```

## DatabaseName (DBN) 接続パラメーター

すでに実行されているデータベースに接続するときに、接続する必要のある、ロードしたデータベースを識別します。

### 構文

{ **DatabaseName** | **DBN** }=*database-name*

### 使用法

実行中のローカルデータベースサーバーかネットワークデータベースサーバー。

### 指定可能な値

- **database-name**　この文字列は、データベースサーバーですでに実行しているデータベースの名前を指定します。

### デフォルト

デフォルト設定はありません。

### 備考

実行されていないデータベースに接続する場合は、DatabaseFile (DBF) パラメーターを使用します。

データベースがサーバーで起動するたびにデータベース名が割り当てられます。これは、管理者が -n オプションを使用して割り当てるか、またはサーバーがファイル名から拡張子とパスを削除した形で割り当てます。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

SQL Anywhere ユーティリティデータベースへの接続に使用できるのは utility_db という名前のデータベースのみです。「ユーティリティデータベース」135 ページを参照してください。

**注意**
データベースの命名には、DatabaseSwitches (DBS) 接続パラメーターで -n オプションを使用するよりも、DatabaseName (DBN) 接続パラメーターを使用することをおすすめします。

接続するデータベースがすでに実行中の場合は、データベースファイルではなくデータベース名を指定してください。

実行中のデータベース名と DatabaseName (DBN) パラメーターで指定した名前が一致した場合にのみ、接続が行われます。

**注意**
データベースファイルとデータベース名の両方を指定すると、指定した名前の実行中のデータベースに接続します (データベースファイルは無視されます)。接続に失敗すると、データベースファイルとデータベース名の両方を使用して、データベースが自動的に起動されます。

**参照**


- 「接続パラメーターと接続文字列」30 ページ
- 「DatabaseName (DBN) プロトコルオプション」379 ページ
- 「DatabaseSwitches (DBS) 接続パラメーター」337 ページ


**例**

*cities.db* という名前のデータベースファイルを起動して、その名前を Kitchener に変更するには、次のコマンドを実行します。

```
dbeng12 cities.db -n Kitchener
```

上記のコマンドを実行したと仮定すると、次に示すコマンドを使用して Kitchener という名前の実行中のデータベースに接続できます。

```
DBN=Kitchener
```

代わりに、次のコマンドを使用して Kitchener という実行中のデータベースに接続することもできます。

```
DBN=Kitchener;DBF=cities.db
```

ただし、次のように指定すると、データベース Kitchener への接続が失敗します。

```
DBF=cities.db
```

# DatabaseSwitches (DBS) 接続パラメーター

データベース起動時に、データベースに指定のオプションを提供します。

### 構文

{ **DatabaseSwitches** | **DBS** }=*database-options*

### 使用方法

データベースが起動されていないときに、データベースサーバーに接続します。この接続パラメーターは、データベースサーバーが実行されていない場合に、指定されたデータベースとオプションでデータベースサーバーを自動的に起動します。

### 指定可能な値

- **データベースオプション**　この文字列は、データベースファイルに適用するオプションを指定します。「データベースオプション (dbeng12/dbsrv12)」305 ページを参照してください。

  データベースサーバーオプションは StartLine 接続パラメーターを使用して指定してください。「StartLine (START) 接続パラメーター」366 ページを参照してください。

### デフォルト

オプションはありません。

### 備考

現在実行していないデータベースに接続するときにのみ、DatabaseSwitches を使用してください。DatabaseFile で指定されたデータベースをサーバーが起動するとき、サーバーは指定された DatabaseSwitches を使用して、データベースの開始オプションを決定します。

> **注意**
> データベースの命名には、DatabaseSwitches (DBS) 接続パラメーターで -n オプションを使用するよりも、DatabaseName (DBN) 接続パラメーターを使用することをおすすめします。

### 参照


- 「SQL Anywhere データベースサーバーの構文」189 ページ
- 「接続パラメーターと接続文字列」30 ページ
- 「DatabaseName (DBN) 接続パラメーター」336 ページ
- 「DatabaseKey (DBKEY) 接続パラメーター」335 ページ


### 例

次のコマンドは、デフォルトのデータベースサーバーに接続し、データベースファイル *demo.db* (DatabaseFile (DBF) 接続パラメーター) をロードし、そのファイルに my-db (DatabaseName (DBN) 接続パラメーター) と名前を付け、これを読み込み専用モード (-r オプション) で開始します。

```
dbisql -c "UID=DBA;PWD=sql;DBF=%SQLANYSAMP12%¥demo.db;DBN=my-db;DBS=-r"
```

## DataSourceName (DSN) 接続パラメーター

ODBC ドライバーマネージャーまたは Embedded SQL ライブラリに対して、レジストリ内またはシステム情報ファイル (デフォルト名は *.odbc.ini*) での ODBC データソース情報の検索場所を指示します。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 構文

{ **DataSourceName** | **DSN** }=*data-source-name*

### 使用法

特に制限なし

### 指定可能な値

- **data-source-name**　この文字列は、データベースの接続情報が含まれる ODBC データソースの名前を指定します。

### デフォルト

デフォルトのデータソース名はありません。

### 備考

データソース名だけを ODBC に送信するのは、ODBC アプリケーションの一般的な手法です。ODBC ドライバーマネージャーと ODBC ドライバーは、接続パラメーターの残りの部分を含んだデータソースを探します。

SQL Anywhere では、Embedded SQL アプリケーションも ODBC データソースを使用して接続パラメーターを保管できます。

### 参照


- 「FileDataSourceName (FILEDSN) 接続パラメーター」344 ページ
- 「接続パラメーターと接続文字列」30 ページ
- 「UNIX での ODBC データソース」48 ページ
- 「ODBC データソース」36 ページ


### 例

次のパラメーターは、データソース名を使用します。

```
DSN=My Database
```

## DisableMultiRowFetch (DMRF) 接続パラメーター

ネットワーク上での複数ローフェッチをオフにします。

### 構文

{ **DisableMultiRowFetch** | **DMRF** }={ **YES** | **NO** }

### 使用法

特に制限なし

### デフォルト

NO

### 備考

デフォルトでは、データベースサーバーが単純なフェッチ要求を受信すると、アプリケーションは追加のローを要求します。このパラメーターを YES に設定すると、この動作を無効にできます。「プロシージャー、トリガー、ユーザー定義関数、バッチのカーソル」『SQL Anywhere サーバー SQL の使用法』を参照してください。

DisableMultiRowFetch (DMRF) 接続パラメーターを YES に設定するのと、prefetch データベースオプションを Off に設定するのとは、同じ効果があります。「プリフェッチ」『SQL Anywhere サーバー プログラミング』を参照してください。

### 参照


- 「接続パラメーターと接続文字列」30 ページ
- 「prefetch オプション」662 ページ


### 例

次の接続文字列フラグメントは、プリフェッチを回避します。

```
DMRF=YES
```

## Elevate 接続パラメーター

Windows Vista で自動的に起動されるデータベースサーバー実行プログラムを昇格します。

### 構文

**Elevate={ YES | NO }**

### 使用法

Windows Vista のみ

### デフォルト

NO

### 備考

接続文字列で ELEVATE=YES を指定すると、自動的に起動されたデータベースサーバー実行プログラムが昇格します。これにより、昇格されていないクライアントプロセスが昇格したサーバーを自動的に起動できるようになります。Windows Vista の場合、昇格されていないサーバーは AWE メモリを使用できないため、この処理が必要です。データベースサーバーが自動的に起動されていない場合、このパラメーターは無視されます。AWE キャッシュを使用するには、データベースサーバーコマンドの実行時に -cw を指定します。

### 参照


- 「Elevate 接続パラメーターの使用」29 ページ
- 「-cm dbeng12/dbsrv12 サーバーオプション」212 ページ
- 「-cw dbeng12/dbsrv12 サーバーオプション (旧式)」217 ページ

### 例

次の接続文字列フラグメントは、Windows Vista で自動的に起動されたデータベースサーバーを昇格して、AWE キャッシュを使用できるようにします。

```
"Elevate=YES;START=dbeng12 -cw"
```

## EncryptedPassword (ENP) 接続パラメーター

パスワードを指定し、データソースに暗号形式で保管します。

### 構文

{ **EncryptedPassword** | **ENP** }=*password*

### 使用法

特に制限なし

### 指定可能な値

- **password**　この文字列は、接続するデータベースの暗号化されたパスワードです。

### デフォルト

なし

### 備考

すべてのデータベースユーザーにはパスワードがあります。パスワードをユーザーに提供し、データベースに接続できるようにしてください。

EncryptedPassword (ENP) 接続パラメーターは、暗号化されたパスワードを指定するために使用します。アプリケーションの接続文字列には、暗号化されたパスワードが含まれることがあります。Password (PWD) 接続パラメーターと EncryptedPassword (ENP) 接続パラメーターの両方を指定した場合、Password (PWD) 接続パラメーターが優先されます。

> **警告**
> データソースを作成するときに、暗号化されたパスワードを定義の一部として含めないことをおすすめします。この機能は Windows **ODBC データ ソース アドミニストレーター**の **[SQL Anywhere の ODBC 設定]** ウィンドウと、SQL Anywhere データソースユーティリティ (dbdsn) の両方にありますが、この情報を含めるとセキュリティに関するリスクが発生します。

データソースユーティリティ (dbdsn) で -pe オプションを使用して、ODBC データソースに格納されたパスワードを暗号化できます。「データソースユーティリティ (dbdsn)」936 ページを参照してください。

UNIX では、データソース情報はシステム情報ファイル (デフォルト名は *.odbc.ini*) に格納されます。このシステム情報ファイルがどのように検索されるかについては、「UNIX での ODBC データソース」48 ページを参照してください。

### 参照


- 「接続パラメーターと接続文字列」30 ページ
- 「Password (PWD) 接続パラメーター」360 ページ


## Encryption (ENC) 接続パラメーター

トランスポートレイヤーセキュリティまたは単純暗号化を使用してクライアントアプリケーションとデータベースサーバー間で送信されるパケットを暗号化します。

### 構文

```
{ Encryption | ENC }={ NONE
 | SIMPLE
 | TLS( TLS_TYPE=algorithm;
   [ FIPS={ Y | N }; ]
  TRUSTED_CERTIFICATE=public-certificate;
   [ CERTIFICATE_COMPANY=organization; ]
   [ CERTIFICATE_NAME=common-name; ]
   [ CERTIFICATE_UNIT=organization-unit ] )
```

### 使用法

TLS の場合：TCP/IP のみサポート

NONE または SIMPLE の場合：特に制限なし

### 指定可能な値

- **NONE**　暗号化されていない通信パケットを受け入れます。

- **SIMPLE**　すべてのプラットフォームと SQL Anywhere の以前のバージョンでサポートされる単純暗号化で暗号化された通信パケットを受け入れます。単純暗号化では、サーバー認証、強力な楕円曲線暗号化、RSA 暗号化、トランスポートレイヤーセキュリティのその他の機能は提供されません。

  データベースサーバーが単純暗号化を受け入れ、暗号化なしを受け入れない場合、暗号化を使用しない TDS 接続以外の接続では、単純暗号化が使用されます。

  -ec SIMPLE を指定してデータベースサーバーを起動すると、データベースサーバーは単純暗号化を使用した接続だけを受け入れます。TLS 接続 (ECC、RSA、RSA FIPS 認定) は失敗し、暗号化を要求しない接続では単純暗号化が使用されます。

  -ec SIMPLE,TLS( TLS_TYPE=ECC;... ) を指定してデータベースサーバーを起動すると、データベースサーバーは ECC TLS 暗号化または単純暗号化を使用する接続だけを受け入れます。RSA 接続と RSA FIPS 接続はいずれも失敗し、暗号化を要求しない接続では単純暗号化が使用されます。

- ***algorithm***　RSA 暗号化の場合は **RSA** を指定し、ECC 暗号化の場合は **ECC** を指定します。FIPS 認定の RSA 暗号化の場合は、**TLS_TYPE=RSA;FIPS=Y** を指定します。RSA FIPS は別の認定ライブラリを使用しますが、SQL Anywhere 9.0.2 以降で RSA を指定しているサーバーと互換性があります。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

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

クライアントは、次の引数を使用して、サーバーのパブリック証明書内のフィールド値を検証できます。

- trusted_certificate
- certificate_company
- certificate_unit
- certificate_name

サーバー認証での証明書のフィールドの検証については、「証明書フィールドの確認」1422 ページを参照してください。

## デフォルト

NONE

## 備考

このパラメーターは、トランスポートレイヤーセキュリティまたは単純暗号化を使用してクライアントアプリケーションとデータベースサーバー間の通信を安全化する場合に使用します。「トランスポートレイヤーセキュリティ」1410 ページを参照してください。

**注意**
別途ライセンスが必要な必須コンポーネント

ECC 暗号化と FIPS 認定の暗号化には、別途ライセンスが必要です。強力な暗号化テクノロジはすべて、輸出規制対象品目です。

「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

デジタル証明書の使用については、「デジタル証明書」1414 ページを参照してください。

CONNECTION_PROPERTY システム関数を使用して、現在の接続の暗号化設定値を取得できます。

```
SELECT CONNECTION_PROPERTY ( 'Encryption' );
```

接続によって使用されている暗号化のタイプに応じて、この関数は None、Simple、ecc_tls、rsa_tls、rsa_tls_fips の 5 つの値のうちいずれか 1 つを返します。

「CONNECTION_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

### 参照


- 「トランスポートレイヤーセキュリティを使用するようにクライアントアプリケーションを設定する」1422 ページ
- 「-ec dbeng12/dbsrv12 サーバーオプション」223 ページ
- 「-ek dbeng12/dbsrv12 データベースオプション」310 ページ
- 「-ep dbeng12/dbsrv12 サーバーオプション」226 ページ
- 「-es dbeng12/dbsrv12 サーバーオプション」227 ページ
- 「接続パラメーターと接続文字列」30 ページ
- 「DatabaseKey (DBKEY) 接続パラメーター」335 ページ
- 「certificate_company プロトコルオプション」374 ページ
- 「certificate_name プロトコルオプション」375 ページ
- 「certificate_unit プロトコルオプション」376 ページ
- 「trusted_certificate プロトコルオプション」401 ページ
- Encryption 接続プロパティ723 ページ


### 例

次の接続文字列フラグメントは、トランスポートレイヤーセキュリティと楕円曲線暗号化を使用して、データベースサーバーに接続します。

```
"Host=myhost;Server=myserver;ENCRYPTION=tls(tls_type=ecc;trusted_certificate=eccroot.crt)"
```

次の接続文字列フラグメントは、トランスポートレイヤーセキュリティと RSA 暗号化を使用して、データベースサーバーに接続します。

```
"Host=myhost;Server=myserver;ENCRYPTION=tls(tls_type=rsa;fips=n;trusted_certificate=rsaroot.crt)"
```

次の接続文字列フラグメントは、単純暗号化を使用して、データベースサーバーに接続します。

```
"Host=myhost;Server=myserver;ENCRYPTION=simple"
```

## EngineName (ENG) 接続パラメーター

この接続パラメーターは使用されなくなりました。代わりに、Use the ServerName (Server) 接続パラメーターを使用します。 「ServerName (Server) 接続パラメーター」365 ページを参照してください。

## FileDataSourceName (FILEDSN) 接続パラメーター

接続するデータベースに関する情報を ODBC ファイルデータソースが保有していることをクライアントライブラリに知らせます。

### 構文

{ **FileDataSourceName** | **FILEDSN** }=*file-data-source-name*

### 使用法

特に制限なし

### 指定可能な値

- **file-data-source-name**　この文字列は、データベースの接続情報が含まれるファイルデータソースの名前を指定します。

### デフォルト

デフォルト名はありません。

### 備考

ファイルデータソースは、レジストリに保管される ODBC データソースと同じ情報を持ちます。ファイルデータソースは、簡単にエンドユーザーに配布できるので、接続情報を各コンピューター上で再構成する必要はありません。

ODBC と Embedded SQL アプリケーションのいずれも、ファイルデータソースを使用できます。

### 参照


- 「Windows でのファイルデータソース」45 ページ
- 「DataSourceName (DSN) 接続パラメーター」338 ページ
- 「接続パラメーターと接続文字列」30 ページ
- 「Windows でのファイルデータソース」45 ページ


## ForceStart (FORCE) 接続パラメーター

実行しているサーバーがあるかどうかをチェックすることなくデータベースを起動します。

### 構文

**{ ForceStart | FORCE }={ YES | NO }**

### 使用法

db_start_engine 関数と併用する場合のみ

### デフォルト

NO

### 備考

ForceStart=YES を設定することで、db_start_engine 機能は、サーバーを起動する前に実行しているサーバーがあるかどうかをチェックしなくなります。

### 参照


- 「db_start_engine 関数」『SQL Anywhere サーバー プログラミング』
- 「接続パラメーターと接続文字列」30 ページ

# Host 接続パラメーター

ホスト名または IP アドレスとオプションのポート番号を受け入れ、データベースサーバーがある場所をクライアントに知らせます。

### 構文

**Host={** *hostname* | *ip-address* **}[ :***port-number* **]** ...

### 使用法

特に制限なし。ネットワークサーバーへの接続には、Host 接続パラメーターをおすすめします (TCP/IP の使用を意味します)。

### 指定可能な値

- **hostname**　データベースサーバーを実行しているコンピューターの名前。ホスト値のリストはカンマで区切られていますが、オプションとしてポート番号を含めることができます (コロンで区切る)。**localhost** を使用して現在のコンピューターを識別できます。

- **ip-address**　この文字列は、IP アドレス形式で指定する必要があり、任意でポート番号を含めることができます (コロンで区切る)。IP アドレスのリストは、カンマで区切られています。

  IPv6 アドレスの場合は、アドレスを角カッコまたはカッコで囲む必要があります。たとえば、8933 がポート番号であるとき、Host=[fd77:ab34:2238::3894]:8933 のように指定します。

- **port-number**　データベースサーバーで使用されるポート番号。デフォルトのポート番号は 2638 です。

### デフォルト

なし

### 備考

Host 接続パラメーターは、1 つ以上のホスト名 (または IP アドレス) とオプションのポート番号を指定し、データベースサーバーが実行している場所をクライアントに知らせます。

Host 接続パラメーターは、データベースサーバーが実行されている 1 つ以上のコンピューターシステムを識別します。データベースサーバーは、Host 接続パラメーターの指定では自動的には起動しません。

コンピューターシステムで 1 つ以上のデータベースサーバーが実行されており、ポートを指定していない場合、ServerName 接続パラメーターを使用して接続するサーバーを指定します。ServerName パラメーターを常に使用することをおすすめします (特に埋め込みアプリケーションの場合)。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| サーバーとポートを指定する | host=myhost:1234;<br>servername=myserver<br><br>指定したポートから受信を行う指定したサーバー名のサーバーのみ接続が確立されます。 |
| サーバーを指定し、ポートは指定しない | host=myhost;<br>servername=myserver<br><br>指定したサーバー名のサーバーとの接続が確立されます。 |
| サーバーもポートも指定しない | host=myhost<br><br>ポート 2638 から受信を行うサーバーに対して接続が試行されます。 |

複数のアドレスが指定された場合、次の順序でアドレスに接続が試行されます。

1. 指定されたアドレスが、データベースサーバーアドレスキャッシュ (sasrv.ini) に対して検査されます。一致するものが見つかった場合には、データベースサーバーキャッシュ内のこのアドレスが最初に試行されます。

2. アドレスは、指定された順序に従って試行されます。

Host パラメーターを使用する場合、サーバーを一意に特定するための十分な情報 (ホスト名とポート番号) が指定されていれば、UDP パケットは送信されません。ポート番号とデータベースサーバー名がどちらも指定されていない場合は、ポート番号は 2638 であると想定され、クライアントはブロードキャストを実行しません。ただし、クライアントにホスト名とデータベースサーバー名があり、ポート番号がない場合は、指定されたホストのポート 2638 に UDP パケットが送信されて、ポート番号が検索されます。「Host 接続パラメーターを使用した SQL Anywhere データベースサーバーの検出」84 ページを参照してください。

HOST プロトコルオプションまたは Host 接続パラメーターと共に LocalOnly プロトコルオプションを指定することはできません。

**注意**
Host 接続パラメーターでは、共有メモリの使用が無効になります。パーソナルサーバーに対して Host 接続パラメーターを使用する場合、サーバーは TCP/IP オプションで起動する必要があります。

CommLinks (LINKS) 接続パラメーターは、HOST または ServerPort (PORT) 以外の TCP/IP プロトコルオプションを指定する必要がある場合のみ使用することをおすすめします。

接続文字列に CommLinks と Host の両方は指定できません。

### 参照


- 「Host (IP) プロトコルオプション (クライアント側のみ)」381 ページ
- 「一般的な接続シナリオ」32 ページ
- 「CommLinks (LINKS) 接続パラメーター」327 ページ
- 「ServerName (Server) 接続パラメーター」365 ページ


### 例

SalesDB という名前のデータベースサーバーが Elora というコンピューターのデフォルトのポート番号で実行されていることがわかっている場合、次の接続文字列を使用して、データベースサーバーに接続できます。

```
UID=DBA;PWD=sql;Server=SalesDB;Host=Elora:2638
```

データベースサーバーが実行されているポート番号が不明の場合は、次の接続文字列を使用して、Elora というコンピューターで実行されている SalesDB データベースサーバーに接続します。

```
UID=DBA;PWD=sql;Server=SalesDB;Host=Elora
```

## Idle 接続パラメーター

接続のアイドルタイムアウト時間を指定します。

### 構文

**Idle=***timeout-value*

### 使用法

TDS 接続以外。他には特に制限なし。TDS 接続 (jConnect を含む) では、SQL Anywhere Idle 接続パラメーターは無視されます。

### 指定可能な値

- **timeout-value**　接続のアイドルタイムアウト時間 (分単位)。Idle 接続パラメーターの最小値は 1 分で、サポートされている最大値は 32767 分です。0 に指定すると、接続に対するアイドルタイムアウトのチェックがオフになります。

### デフォルト

240 分 (TCP/IP の場合)

0 (共有メモリの場合)

### 備考

Idle 接続パラメーターは、現在の接続に対してのみ適用されます。同一データベースサーバー上の複数の接続に異なるタイムアウト値を設定できます。

Idle 接続パラメーターが使用されていない場合、TCP/IP 接続のアイドルタイムアウト値は -ti データベースサーバーオプションによって制御されます。両方の -ti データベースサーバーオプションと Idle 接続パラメーターが指定された場合、アイドルタイムアウト値は接続パラメーターによって制御されます。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照


- 「-ti dbeng12/dbsrv12 サーバーオプション」277 ページ
- 「接続パラメーターと接続文字列」30 ページ
- 「タイムアウト値の調整」1215 ページ
- IdleTimeout 接続プロパティ726 ページ


### 例

次の接続文字列フラグメントは、この接続のタイムアウト値を 10 分に設定します。

```
"Host=myhost;IDLE=10"
```

## Integrated (INT) 接続パラメーター

統合化ログインを試行できるかどうかを指定します。

### 構文

**{ Integrated | INT }={ YES | NO }**

### 使用法

特に制限なし

### 指定可能な値

- **YES**　統合化ログインを行います。接続の試行が失敗し、login_mode オプションが Standard、Integrated に設定されている場合、標準ログインを試行します。
- **NO**　これがデフォルト設定です。統合化ログインは試行されません。

### デフォルト

NO

### 備考

統合化ログインを使用するクライアントアプリケーションでは、login_mode データベースオプションを Integrated に設定してサーバーを実行してください。

### 参照


- 「login_mode オプション」635 ページ
- 「接続パラメーターと接続文字列」30 ページ
- 「Windows 統合ログイン」51 ページ


### 例

次のデータソースフラグメントは、統合化ログインを使用します。

```
INT=YES
```

# Kerberos (KRB) 接続パラメーター

データベースサーバーへの接続時に Kerberos 認証を使用できるかどうかを指定します。

### 構文

{ **Kerberos** | **KRB** }= { **YES** | **NO** | **SSPI** | *GSS-API-library-file* }

### 使用法

Windows Mobile を除くすべてのプラットフォーム

### 指定可能な値

- **YES**　Kerberos 認証ログインが試行されます。
- **NO**　Kerberos 認証ログインは試行されません。これはデフォルトです。
- **SSPI**　Kerberos 認証ログインが試行され、GSS-API ライブラリの代わりに組み込みの Windows SSPI インターフェイスが使用されます。SSPI は Windows プラットフォームでのみ使用できます。また、Domain Controller Active Directory KDC 以外のキー配布センター (KDC) で SSPI を使用することはできません。Windows クライアントコンピューターがすでに Windows ドメインにログインされている場合は、Kerberos クライアントをインストールまたは設定せずに、SSPI を使用できます。

  **注意**
  SSPI は、SQL Anywhere クライアントのみが Kerberos 接続パラメーターで使用できます。SQL Anywhere データベースサーバーは SSPI を使用できません。サポートされる SSPI 以外の Kerberos クライアントを使用する必要があります。

- **GSS-API-library-file**　Kerberos 認証ログインが試行されます。この文字列は、Kerberos GSS-API ライブラリ (または UNIX の共有オブジェクト) のファイル名を指定します。これは、Kerberos クライアントでデフォルトと異なる Kerberos GSS-API ライブラリファイル名が使用されているか、コンピューターに複数の GSS-API ライブラリがインストールされている場合にだけ必要です。

### デフォルト

NO

### 備考

Kerberos 認証ログインを使用している場合、UserID および Password 接続パラメーターは無視されます。

Kerberos 認証を使用するには、Kerberos クライアントがインストールおよび設定され (SSPI の場合は不要)、ユーザーが Kerberos にログインされ (有効なチケット付与権限付きチケットがある)、データベースサーバーで Kerberos 認証ログインが有効になり設定されている必要があります。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照

- 「-kl dbeng12/dbsrv12 サーバーオプション」250 ページ
- 「-kr dbeng12/dbsrv12 サーバーオプション (旧式)」252 ページ
- 「-krb dbeng12/dbsrv12 サーバーオプション」253 ページ
- 「Kerberos 認証」61 ページ
- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「Windows で Kerberos ログインに SSPI を使用する」68 ページ

### 例

```
Kerberos=YES
Kerberos=SSPI
Kerberos=c:¥Program Files¥MIT¥Kerberos¥bin¥gssapi32.dll
```

## Language (LANG) 接続パラメーター

接続の言語を指定します。

### 構文

{ **Language** | **LANG** }=*language-code*

### 使用法

特に制限なし

### 指定可能な値

- **language-code**　2 文字の組み合わせで言語を表します。たとえば、LANG=DE の場合、デフォルト言語をドイツ語に設定します。

  言語コードの詳細については、「ロケール言語」485 ページを参照してください。

### デフォルト

SALANG 環境変数、dblang ユーティリティ、インストーラーの順に指定された言語

### 備考

この接続パラメーターは接続用の言語を設定するものです。サーバーが指定された言語をサポートしている場合に、サーバーからのエラーや警告が指定された言語で配信されます。

言語を指定しない場合は、デフォルトの言語が使用されます。デフォルトの言語は、SALANG 環境変数、dblang ユーティリティ、インストーラーの順で指定された言語です。

この接続パラメーターは接続のみに影響します。SQL Anywhere ツールとユーティリティから返されるメッセージはデフォルトの言語で表示されますが、サーバーから戻されるメッセージは接続の言語で表示されます。

### 参照

- 「接続パラメーターと接続文字列」30 ページ

# LazyClose (LCLOSE) 接続パラメーター

カーソル要求を次の要求が発生するまでキューイングするか、またはただちに実行するかを制御します。カーソルを閉じる要求をキューイングすると、往復がなくなり、パフォーマンスが向上します。

### 構文

**{ LazyClose | LCLOSE }={ YES | NO | AUTO }**

### 使用法

特に制限なし

### 指定可能な値

- **YES**　カーソルを閉じる要求を常にキューイングするため往復がなくなりますが、クライアントによってカーソルが閉じられた後に、ロックが発生したり、他のリソースが保留状態になる場合があります。同じ接続でデータベースサーバーに次の要求が送信されると、カーソルを閉じる要求が実行されます。CLOSE cursor-name データベース要求がキューイングされている間は、独立性レベル 1 のすべてのカーソル安定性ロックがカーソルに適用されます。

- **NO**　カーソルをただちに閉じます。

- **AUTO**　ロックや大量のサーバーリソースの保留状態が持続する時間が変わらない場合のみ、カーソルを閉じる要求をキューイングして往復をなくします。カーソルが独立性レベル 1 のカーソル安定性ロックを使用する場合、またはカーソルが閉じられるまでサーバーリソースが解放されないために大量のリソースを消費する可能性がある場合、カーソルはただちに閉じられます。ワークテーブルを必要とするクエリなどは、大量のサーバーリソースを消費する可能性の高いカーソルの例です。

### デフォルト

AUTO

### 備考

この接続パラメーターを YES または AUTO に設定すると、カーソルは次のデータベース要求が発生するまで閉じられません。

ネットワークの遅延時間が長い場合、またはアプリケーションでカーソルを開く要求と閉じる要求を多数送信する場合は、このオプションを有効にすると、パフォーマンスが向上します。

### 参照


- 「接続パラメーターと接続文字列」30 ページ
- 「クライアントとサーバーとの間の要求数の削減」『SQL Anywhere サーバー SQL の使用法』


# LivenessTimeout (LTO) 接続パラメーター

不完全な接続の終了を制御します。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 構文

{ **LivenessTimeout** | **LTO** }=*timeout-value*

### 使用法

ネットワークサーバーのみ

非スレッド化 UNIX アプリケーションを除くすべてのプラットフォーム

### 指定可能な値

- **timeout-value**　接続の活性タイムアウト期間 (秒単位)。LivenessTimeout 接続パラメーターの最小値は 30 秒で、最大値は 32767 秒です。0 に指定すると、接続に対する活性タイムアウトのチェックがオフになります。最小値より小さい 0 以外の値は最小値に再設定されます。たとえば、"LivenessTimeout=5" が含まれている接続文字列を指定すると、"LivenessTimeout=30" が使用されます。LivenessTimeout 値が設定されていない場合、LivenessTimeout はサーバー設定で制御されます。デフォルトは 120 秒です。

### デフォルト

なし

### 備考

接続が維持されていることを確認するため、クライアント／サーバーの TCP/IP 通信プロトコルを介して、定期的に「活性パケット」が送信されます。活性要求や応答パケットを検出することなく、指定した LivenessTimeout 期間にわたってクライアントが実行されていると、通信は切断されます。

LivenessTimeout 値の 3 分の 1 から 3 分の 2 の期間で接続がパケットを送信しない場合に、活性パケットが送信されます。

サーバーへの接続数が 200 を超えると、サーバーは指定された LivenessTimeout 値に基づいて、それより大きい LivenessTimeout 値を自動的に算出します。これにより、サーバーはより多くの接続を効率よく処理できます。

またこのパラメーターは、**[SQL Anywhere の ODBC 設定]** ウィンドウの **[詳細]** タブで、**[活性タイムアウト]** テキストボックスに値を入力して設定することもできます。

### 参照


- 「接続パラメーターと接続文字列」30 ページ
- 「-tl dbeng12/dbsrv12 サーバーオプション」278 ページ
- LivenessTimeout 接続プロパティ 728 ページ


### 例

次の接続文字列フラグメントでは、LivenessTimeout 値を 10 分に設定します。

```
LTO=600
```

# LogFile (LOG) 接続パラメーター

ファイルにクライアントエラーメッセージとデバッグメッセージを送信します。

### 構文

{ **LogFile** | **LOG** }=*filename*

### 使用法

特に制限なし

### 指定可能な値

- **filename**　この文字列は、クライアントエラーメッセージとデバッグメッセージが保存されるファイルの名前を指定します。ファイル名にパスがない場合、クライアントアプリケーションの現在の作業ディレクトリを基準にします。

### デフォルト

ログファイルはありません。

### 備考

LogFile (LOG) 接続パラメーターは接続ごとに固有であるため、単一のアプリケーションで、接続ごとに異なる LogFile 引数を設定できます。

一般的なログファイルの内容は次のとおりです。

```
Mon Jan 17 2011 08:17:56
08:17:56
UID=DBA;PWD=********;DBF='C:¥Documents and Settings¥All Users¥Documents¥SQL Anywhere
12¥Samples¥demo.db';ServerName=demo12;START='C:¥Program Files¥SQL Anywhere
12¥Bin32¥dbeng12.exe';CON=SQL_DBC_47d853e8;ASTOP=YES;LOG=c:¥logs¥t を使用して接続を試
みています。
08:17:56 動作中のサーバーへの接続を試みています...
08:17:56 SharedMemory リンクを開始しています...
08:17:56    SharedMemory リンクは正常に開始しました。
08:17:56 SharedMemory 接続を試みています (sasrv.ini キャッシュアドレスがありません)。
08:17:56 SharedMemory によってサーバーに接続しました。
08:17:56 バージョン 12.0.1.3149 の SQL Anywhere サーバーに接続しました。
08:17:56 アプリケーション情報 :
08:17:56 IP=10.7.169.107;HOST=MCBRIDEM-PC;OSUSER=user1;OS='Windows XP Build 2600
Service Pack 2';EXE='C:¥Program Files¥SQL Anywhere
12¥bin32¥dbisql.exe';PID=0xa4c;THREAD=0x1430;VERSION=12.0.1.3149;API=iAnywhereJDBC;TIME
ZONEADJUSTMENT=-480
08:17:56 サーバーに接続しました。実行中のデータベースへの接続を試みています...
08:17:56 [    2] データベースに正常に接続しました。
08:17:57 [    2] プリフェッチバッファーのために、プリフェッチローの数が 16 に減りました。
08:17:57 [    2] 制限値に達しました。PrefetchBuffer 接続パラメーターを増やすと、パフォーマンスが向
上する場合があります。
```

### 参照


- 「接続パラメーターと接続文字列」30 ページ
- 「LogFile (LOG) プロトコルオプション」388 ページ



Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 例

次のコマンドラインは、LogFile (LOG) 接続パラメーターを使用して、Interactive SQL を起動して、サンプルデータベースに接続します。

```
dbisql -c "DSN=SQL Anywhere 12 Demo;LOG=c:¥logs¥test.txt"
```

## NewPassword (NEWPWD) 接続パラメーター

パスワードの有効期限が切れている場合でも、DBA の介入を必要とせずユーザーが各自でパスワードを変更できます。

### 構文

{ **NewPassword** | **NEWPWD** }={ *password-string* | * }

### 使用法

特に制限なし。新しいパスワードの入力を要求するクライアントライブラリは、Microsoft Windows でのみサポートされます。

### 指定可能な値

- **password-string**　ユーザーが新しいパスワードを入力すると、データベースサーバーはユーザー ID とパスワードの認証を実行し、login_procedure オプションが呼び出される前にパスワードを変更しようとします。このプロセスにより、ユーザーは DBA の介入なく有効期限の切れたパスワードを変更できるようになります。verify_password_function オプションを設定した場合は、新しいパスワードの検証が行われます。統合化ログインや Kerberos ログインで認証を実行する場合、元のパスワードは検証されず、データベースサーバーは新しいパスワード値を無視し、パスワードは変更されません。

- *****　Microsoft Windows の場合、特別値 * を使用すると、既存のパスワードの有効期限が切れている場合にかぎり、クライアントライブラリは新しいパスワードの入力を要求します。ユーザーは既存のパスワードと新しいパスワードを入力し、さらに確認のために新しいパスワードをもう一度入力する必要があります。ユーザーがフィールドに値を入力して **[OK]** をクリックすると、古いパスワードが認証され、データベースサーバーはパスワードを変更しようとします。verify_password_function オプションを設定した場合は、新しいパスワードの検証が行われます。ユーザーのパスワードの有効期限切れを確認し、新しいパスワードの入力を要求し、パスワードの認証と変更を行うプロセスは、クライアントライブラリに対する 1 回の接続呼び出しで実行されます。

### デフォルト

パスワードは変更されず、クライアントライブラリは新しいパスワードを要求しません。

### 備考

この接続パラメーターは、password_life_time オプションや password_expiry_on_next_login オプションを使用してログインポリシーを実装する場合に非常に便利です。また、login_procedure で「**パスワードの有効期限が切れています。**」というエラーを伝えることで、パスワードの有効期限ポリシーを実装することもできます。

動作環境でパスワードの入力要求がサポートされていない場合、ユーザーは「**パスワードの有効期限が切れています。**」というエラーを受信します。Microsoft Windows 環境の場合、呼び出し元のアプリケーションに最上位レベルのウィンドウが複数あったり、最小化されている場合、プロンプトウィンドウでは呼び出し元のアプリケーションウィンドウとの対話を正しく防げない (モーダルでなくなったり、正しい親ウィンドウがない) ことがあります。

Windows 環境では、ODBC SQLDriverConnect 関数を使用していて DriverCompletion 引数が SQL_DRIVER_NOPROMPT 以外に設定されている場合、パスワードの有効期限が切れていると、接続時に新しいパスワードの入力を要求されます。DBPROP_INIT_PROMPT プロパティが DBPROMPT_NOPROMPT 以外に設定されている場合、OLE DB で接続時に新しいパスワードの入力を要求されることがあります。いずれの場合も、接続パラメーター NewPassword=* が指定されているように動作します。

●ユーザー ID には、次に該当する値を指定できません。

○ 空白スペース、一重引用符、または二重引用符で始まる値
○ 空白スペースで終わる値
○ セミコロンを含む値

●パスワードは大文字と小文字が区別され、次に該当する値は指定できません。

○ 空白スペース、一重引用符、または二重引用符で始まる値
○ 空白スペースで終わる値
○ セミコロンを含む値
○ 256 バイト長を超える値

**参照**


- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「login_procedure オプション」636 ページ
- 「verify_password_function オプション」708 ページ
- 「post_login_procedure オプション」659 ページ


**例**

次の接続文字列では、接続時にユーザー Test1 のパスワードを変更します。

```
"UID=Test1;PWD=welcome;NEWPWD=hello"
```

Windows 環境の場合、次の接続文字列では、既存のパスワードの有効期限が切れている場合に、ユーザー Test1 に新しいパスワードの入力を要求します。

```
"UID=Test1;PWD=welcome;NEWPWD=*"
```

# MatView (MATVIEW) 接続パラメーター

ODBC 関数の SQLTables が呼び出されたときに、マテリアライズドビューに返される文字列を指定します。

### 構文

**MATVIEW=***string*

### 使用法

ODBC

### デフォルト

VIEW

### 備考

デフォルトでは、ODBC 関数 SQLTables を実行したときに、値 VIEW がマテリアライズドビューの TABLE_TYPE カラムに返されます。異なる戻り値を指定するには、MatView 接続パラメーターを使用します。指定する文字列は、SQLTables 関数が TableType フィルターで呼び出されるときにも返されます。

### 参照

- 「ODBC サポート」『SQL Anywhere サーバー プログラミング』

### 例

SQLTables() 関数に、タイプ MATERIALIZED VIEW としてマテリアライズドビューを返させるには、次のように接続します。

```
UID=DBA;PWD=sql;ServerName=server-name;MATVIEW=MATERIALIZED VIEW
```

## NodeType (NODE) 接続パラメーター

接続パラメーターで最終的に接続されるサーバーの役割を制御します。高可用性設定、読み込み専用設定、スケールアウト設定、クラウド設定で使用します。設定によっては、接続が他のサーバーにリダイレクトされ、負荷分散機能も提供されます。

> **注意**
> 読み込み専用のスケールアウトとデータベースミラーリングには、それぞれ別のライセンスが必要です。 「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

### 構文

**{ NodeType | NODE }={ DIRECT | PRIMARY | MIRROR | COPY | READONLY}**

### 使用方法

ネットワークサーバーのみ

### 指定可能な値

- **DIRECT** これは、クラウドデータベースでないサーバーに接続する場合のデフォルト設定です。

NodeType 接続パラメーターが DIRECT に設定されている場合、データベースサーバーは、負荷分散またはリダイレクションを行わずに接続を受け入れます。

- **PRIMARY** これは、クラウドサーバーに接続する場合のデフォルト設定です。

  プライマリサーバーに接続したときに、NodeType 接続パラメーターが PRIMARY に設定されている場合、接続は受け入れられます。プライマリーでないサーバー (ミラーサーバー、コピーノード、その他のクラウドデータベース) に接続した場合、データベースサーバーは接続をプライマリサーバーにリダイレクトします。

  この NodeType を使用してクラウドサーバーに接続するときは、ServerName 接続パラメーターを指定しません。

- **COPY** NodeType 接続パラメーターを COPY に設定した場合、データベースサーバーは負荷分散を実行してコピーノードを選択します。クラウドでは、コピーノードが無作為に選択されます。クラウドでない読み込み専用のスケールアウトシステムでは、データベースサーバーはその分岐内のコピーノード (ルートノードでない場合はサーバー自体も) を検査し、負荷が最も軽いコピーノードを選択します。データベースサーバーがサーバー自体を選択しない場合、選択したデータベースサーバーにクライアントをリダイレクトします。

  この NodeType を使用してクラウドサーバーに接続するときは、ServerName 接続パラメーターを指定しません。

- **MIRROR** ミラーサーバーに接続したときに、NodeType 接続パラメーターが MIRROR に設定されている場合、接続は受け入れられます。ミラーでないサーバーに接続した場合、データベースサーバーは接続をミラーサーバーにリダイレクトします。

  この NodeType を使用してクラウドサーバーに接続するときは、ServerName 接続パラメーターを指定しません。

- **READONLY** NodeType 接続パラメーターを READONLY に設定した場合、コピーノードまたはミラーの読み込み専用サーバーに接続されます。コピーノードがない場合は、これは MIRROR 設定と同様になります。コピーノードがある場合は、これは COPY 設定と同様になります。

  この NodeType を使用してクラウドサーバーに接続するときは、ServerName 接続パラメーターを指定しません。

### デフォルト

DIRECT (クラウドでないデータベースサーバー)

PRIMARY (クラウドデータベースサーバー)

### 備考

この接続パラメーターは、クライアントアプリケーションが接続するサーバーの役割と、読み込み専用のスケールアウトに関係するデータベースサーバー間の負荷分散を、クライアントアプリケーションで制御するために使用します。クライアントはこの接続パラメーターを指定して、接続するデータベースサーバーのタイプを指示します。クライアントが最初に接続されたデータベースサーバーでは、どのデータベースサーバーが接続を処理するかが判断され、必要に応じ

て、データベースサーバーのアドレスを返すことによって、適切なデータベースサーバーに接続がリダイレクトされます。クライアントは元のデータベースサーバーから自動的に切断され、適切なデータベースサーバーに自動的に接続されます。

**クラウドでないデータベースサーバーで NodeType を使用する**　COPY を指定すると、クライアントが最初に接続されたデータベースサーバーが含まれる分岐で負荷分散が発生します。分岐は、そのデータベースサーバーとその子で構成されます。この機能は、地理的に離れているサーバー間で負荷分散を実行する場合に便利です。

クライアント接続が異なるデータベースサーバーにリダイレクトされると、ServerName (Server) 接続パラメーターで指定された値と Name データベースサーバープロパティの値は一致しません。Name サーバープロパティ 766 ページを参照してください。

NodeType が指定されていると、通常、アプリケーションはルートノードに接続され、負荷が最も低いコピーノードがデータベースサーバーによって決まります。その時点で、そのコピーノードに接続がリダイレクトされます。このような接続と切断をアプリケーションで短時間に何度も行うと、接続がプールされ、どのコピーサーバーを使用するかがルートノードに問われなくなります。接続プールを使用すると、ルートノードでの負荷は削減されますが、予期する結果にならない可能性があります。接続のたびに接続先となるコピーノードをルートサーバーで確実に判断するため、接続のプールを禁止するようアプリケーションで指定することができます。

このパラメーターを使用して Kerberos 認証または統合化ログインを使用する接続をリダイレクトしている場合、スケールアウトツリーのすべてのデータベースサーバーを次のように設定する必要があります。

○ **統合化ログイン**　スケールアウトツリーのすべてのデータベースサーバーは、同一の Windows ユーザーグループの一部であるか、同一の Windows ユーザーがアクセスすることを許可するように設定する必要があります。

○ **Kerberos 認証**　スケールアウトツリーのすべてのデータベースサーバーでは、Kerberos 認証を有効にしておく必要があります。スケールアウトツリーのすべてのサーバーに対する Kerberos サーバープリンシパルは、同一の領域を持つ必要があります。

## 参照


- 「データベースミラーリング」1137 ページ
- 「SQL Anywhere の読み込み専用のスケールアウト」1188 ページ
- 「ServerName (Server) 接続パラメーター」365 ページ


## 例

- プライマリサーバーとミラーサーバーは、Ontario 州 Waterloo にあります。日本で実行されているクライアントは、接続の遅延を削減するため、日本にあるコピーノードのグループを設定します。日本のコピーノードの分岐がルートノードの直接の子である場合は、日本のコピーノードの下に他の子を追加できます。クライアントは、ハブに接続し、NodeType=COPY を指定します。日本にあるローカルの読み込み専用接続は、ローカルデータベースサーバー間で負荷分散され、クライアントは他の場所にあるデータベースサーバーに接続されなくなります。クライアントでデータベースに変更を加える必要がある場合は、接続文字列を変更して NodeType=PRIMARY を使用します。

●クラウドデータベース設定で実行されており、kangaroo1 と kangaroo2 の 2 つのホストがクラウドサーバーを実行している、joey という名前の読み込み／書き込みデータベースに接続するには、次の接続パラメーターを使用します。

```
UID=rick;PWD=secret;DBN=joey;Host=kangaroo1,kangaroo2
```

●クラウドデータベース設定で実行されており、kangaroo1 と kangaroo2 の 2 つのホストがクラウドサーバーを実行している、joey という名前の読み込み専用データベースに接続するには、次の接続パラメーターを使用します。

```
UID=rick;PWD=secret;DBN=joey;Host=kangaroo1,kangaroo2;NodeType=READONLY
```

# Password (PWD) 接続パラメーター

接続時のパスワードを指定します。

### 構文

{ **Password** | **PWD** }=*password-string*

### 使用法

特に制限なし

### 指定可能な値

● **password-string**　パスワードは最大長が 255 バイトで、大文字と小文字が区別されます。パスワードには、前後のスペースやセミコロンを含めることができます。

### デフォルト

パスワードの指定なし

### 備考

すべてのデータベースユーザーにはパスワードがあります。パスワードをユーザーに提供し、データベースへの接続が許可されるようにしてください。

●ユーザー ID には、次に該当する値を指定できません。

○ 空白スペース、一重引用符、または二重引用符で始まる値
○ 空白スペースで終わる値
○ セミコロンを含む値

●パスワードは大文字と小文字が区別され、次に該当する値は指定できません。

○ 空白スペース、一重引用符、または二重引用符で始まる値
○ 空白スペースで終わる値
○ セミコロンを含む値
○ 256 バイト長を超える値

Password (PWD) 接続パラメーターは暗号化されていません。アプリケーションには、パスワードが接続文字列に含まれることがあります。Password (PWD) 接続パラメーターと

EncryptedPassword (ENP) 接続パラメーターの両方を指定した場合、Password (PWD) 接続パラメーターが優先されます。

**警告**
データソースを作成するときに、パスワードを定義の一部として含めないことをおすすめします。この機能は Windows **ODBC データ ソース アドミニストレーター**の **[SQL Anywhere の ODBC 設定]** ウィンドウと、SQL Anywhere データソースユーティリティ (dbdsn) の両方にありますが、この情報を含めるとセキュリティに関するリスクが発生します。

UNIX では、データソース情報はシステム情報ファイル (デフォルト名は *.odbc.ini*) に格納されます。システム情報ファイルがどのように検索されるかについては、「UNIX での ODBC データソース」48 ページを参照してください。

### 参照


- 「Userid (UID) 接続パラメーター」368 ページ
- 「パスワード」515 ページ
- 「パスワードのセキュリティ」1383 ページ
- 「EncryptedPassword (ENP) 接続パラメーター」341 ページ
- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「大文字と小文字の区別」『SQL Anywhere サーバー SQL の使用法』
- 「接続パラメーターと接続文字列」30 ページ


### 例

次の接続文字列フラグメントは、ユーザー ID DBA とパスワード sql を指定します。

```
UID=DBA;PWD=sql
```

## PrefetchBuffer (PBUF) 接続パラメーター

ローをバッファーするためのメモリの最大容量を、バイト単位で設定します。

### 構文

{ **PrefetchBuffer** | **PBUF** }=*buffer-size*[ **k** | **m** ]

### 使用法

特に制限なし

### 指定可能な値

- **buffer-size**　デフォルト値はバイト単位ですが、**k** または **m** を使用してキロバイトまたはメガバイトの単位を指定できます。この接続パラメーターには、64 KB ～ 8 MB の値を指定できます。

  以前のバージョンとの互換性を保つため、16384 未満の値が指定されるとキロバイト単位だと解釈されます。

PrefetchBuffer 接続パラメーターにおいて、k サフィックスを使用しないキロバイト単位での指定は廃止されました。 「PrefetchRows (PROWS) 接続パラメーター」363 ページを参照してください。

### デフォルト

512 KB (524288) (Windows Mobile を除くすべてのプラットフォーム)

64 KB (65536 バイト) (Windows Mobile)

### 備考

PrefetchBuffer (PBUF) 接続パラメーターは、プリフェッチされたローを格納するためにクライアントで割り付けられる接続ごとの最大メモリを制御します。

状況によっては、プリフェッチされるローの数を増やすと、クエリのパフォーマンスが向上することがあります。プリフェッチされるローの数は、PrefetchRows (PROWS) と PrefetchBuffer (PBUF) 接続パラメーターを使用して増やすことができます。

PrefetchBuffer (PBUF) 接続パラメーターを増やすと、GET DATA 要求のバッファーに使用できるメモリ容量も増えます。多数の GET DATA (SQLGetData) 要求を処理するアプリケーションでは、このように設定するとパフォーマンスが向上します。

### 参照

- 「接続パラメーターと接続文字列」30 ページ

### 例

次の接続文字列フラグメントを使用して、PrefetchBuffer によるメモリ制限によってプリフェッチされるローの数が減っているかどうかを判断できます。

```
...PrefetchRows=100;LogFile=c:¥client.txt
```

次の文字列を使用して、メモリ制限を 2 MB に増やすことができます。

```
...PrefetchRows=100;PrefetchBuffer=2M
```

## PrefetchOnOpen 接続パラメーター

パラメーターが有効であるときに、カーソルを開く要求を含むプリフェッチ要求が送信されます。

### 構文

**PrefetchOnOpen={ YES | NO }**

### 使用法

ODBC

### デフォルト

NO

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 備考

このオプションを有効にすると、カーソルを開く要求を含むプリフェッチ要求が送信されます。これにより、カーソルを開くたびにローをフェッチするネットワーク要求は行われなくなります。カーソルを開くときにプリフェッチを実行するには、カラムをバインドしておきます。PrefetchOnOpen を使用するときに、カーソルを開いた後で最初のフェッチの前にカラムを再バインドすると、パフォーマンスが低下することがあります。

結果セットを返すクエリまたはストアドプロシージャーで ODBC の SQLExecute または SQLExecDirect を呼び出すと、カーソルが開きます。

次の場合は、このオプションを有効にするとパフォーマンスが向上します。

●ネットワークの遅延時間が長い。

●アプリケーションがカーソルを開く要求と閉じる要求を多数送信する。

## PrefetchRows (PROWS) 接続パラメーター

データベースのクエリ時にプリフェッチされる、最初に推奨されるローの数を指定します。

### 構文

{ **PrefetchRows** | **PROWS** }=*number-of-rows*

### 使用法

特に制限なし

### 指定可能な値

● **number-of-rows**　プリフェッチされるローの数は、PrefetchRows (PROWS) 接続パラメーターと PrefetchBuffer (PBUF) 接続パラメーターの両方によって制限されており、そのため、プリフェッチされたローの格納に使用できるメモリが制限されます。「PrefetchBuffer (PBUF) 接続パラメーター」361 ページを参照してください。

プリフェッチできるロー数の最大値は 10000 です。

### デフォルト

10

ADO.NET の場合は 200

### 備考

デフォルトでは、プリフェッチローの数を増やすことによってパフォーマンスが向上するカーソルに対して、プリフェッチローの数がクライアントライブラリで動的に増加されます。この動作のため、PrefetchRow のデフォルトを増加しても、ほとんどのアプリケーションではパフォーマンスがあまり向上しません。

PrefetchRow のデフォルトを増加することによってパフォーマンスが向上する可能性があるのは、STATIC または FORWARD ONLY FOR UPDATE カーソルタイプを使用していてもほとんど

の場合には次の操作をフェッチし、位置付け更新や削除をほとんど行わない ODBC、OLE DB、SQL Anywhere JDBC アプリケーションの場合であり、特にクライアントとデータベースサーバーが低速ネットワークまたは広域ネットワークを介して通信している場合です。

プリフェッチされるローの数は、プリフェッチされるローの保管に使用できるメモリ量を制限する PrefetchBuffer (PBUF) 接続パラメーターによって制限されます。「PrefetchBuffer (PBUF) 接続パラメーター」361 ページを参照してください。

プリフェッチできるロー数の最大値は 10000 です。

### 参照

● 「接続パラメーターと接続文字列」30 ページ

### 例

次の接続文字列フラグメントは、プリフェッチされるローの数を 100 に設定します。

```
...PrefetchRows=100;...
```

## RetryConnectionTimeout (RetryConnTO) 接続パラメーター

クライアントライブラリ (DBLIB、ODBC、ADO など) に、サーバーが見つからないうちは指定した時間が経過するまで接続を試行し続けるよう指示します。

### 構文

{ **RetryConnectionTimeout** | **RetryConnTO** }=*timeout-value*

### 使用法

特に制限なし

### 指定可能な値

● **timeout-value**　この接続パラメーターで指定される値は、秒単位のタイムアウトです。接続試行のリトライ回数を示すカウンター値ではありません。デフォルト値の 0 は、接続を 1 度だけ試行することを示します。

### デフォルト

0

### 備考

間隔は 0.5 秒で、接続しようとしてデータベースサーバーが見つからなかった場合にだけリトライされます。それ以外のエラーは、すぐに返されます。データベースサーバーが見つからなかった場合、少なくとも RetryConnectionTimeout 接続パラメーターに指定されている期間、接続が試行されます。

デフォルトの TCP タイムアウトは 5 秒です。接続文字列に 5 秒未満に指定された RetryConnTO が含まれている場合でも (Host=host-name;RetryConnTO=3 など)、5 秒間は接続が試行されます。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照

- 「Timeout (TO) プロトコルオプション」400 ページ

### 例

次の接続文字列フラグメントは、接続試行を最低 5 秒間リトライし続けるようクライアントライブラリに指示します。

```
...RetryConnTO=5;...
```

## ServerName (Server) 接続パラメーター

接続する実行中のデータベースサーバー名を指定します。

### 構文

{ **ServerName** | **Server** }=*database-server-name-string*

### 使用法

ネットワークサーバーまたはパーソナルサーバー

### 指定可能な値

- **database-server-name-string**　データベースサーバーを自動的に起動する場合、このパラメーターを使ってサーバー名を指定できます。

  サーバー名は、クライアントコンピューターの文字セットに従って解釈されます。サーバー名に非 ASCII 文字を使用することは推奨できません。

  この名前は、有効な識別子であることが必要です。データベースサーバー名には、次に該当する値を指定できません。

  - 空白スペース、一重引用符、または二重引用符で始まる値
  - 空白スペースで終わる値
  - セミコロン、フォワードスラッシュ (/)、またはバックスラッシュ (¥) を含む値
  - 長さが 250 バイトを超える値
  - UNIX 上で実行するときにスペースを含む値

  データベースサーバーのロングネームは、プロトコルごとに異なる長さにトランケートされます。

  Windows と UNIX では、データベースサーバーがバージョン 10.0.0 以降で、名前が次の長さを超えている場合、バージョン 9.0.2 以前のクライアントから接続することはできません。

  - Windows 共有メモリの場合は、40 バイト
  - UNIX 共有メモリの場合は、31 バイト
  - TCP/IP の場合は、40 バイト

### デフォルト

デフォルトのローカルデータベースサーバー

### 備考

EngineName と ENG は、下位互換性を保つために受け入れられますが、廃止されました。

データベースサーバーは起動するとき、そのコンピューターのデフォルトのデータベースサーバーになろうとします。デフォルトのサーバーがない場合、最初に起動したデータベースサーバーが、デフォルトのデータベースサーバーになります。そのコンピューターで、データベースサーバー名を明示的に指定しないで共有メモリに接続しようとすると、デフォルトのサーバーに接続されます。

デフォルトのローカルデータベースサーバーに接続する場合は、ServerName は必要ありません。

TCP/IP を介して接続し、Host 接続パラメーターを使用する場合、ホスト名が指定されていれば ServerName は必要ありません。

**[接続]** ウィンドウや **[SQL Anywhere の ODBC 設定]** ウィンドウの場合、**[サーバー名]** フィールドがこれに相当します。

**注意**
配備されたアプリケーションの接続文字列には、ServerName パラメーターを含めることをおすすめします。これにより、コンピューターで複数の SQL Anywhere データベースサーバーが実行されている場合に、アプリケーションが確実に正しいサーバーに接続できるため、タイミングに依存する接続エラーを防ぐことができます。

配備されたアプリケーションで使用されているデータベースサーバーには -xd オプションを使用すること、またすべてのクライアントが、Server 接続パラメーターを使用して、接続先のデータベースサーバー名を明示的に指定することをおすすめします。このようにすると、コンピューターで複数の SQL Anywhere データベースサーバーが実行されているときに、データベースが正しいデータベースサーバーに接続します。

### 参照


- 「ServerName (Server) 接続パラメーターの使用」28 ページ
- 「識別子」『SQL Anywhere サーバー SQL リファレンス』
- 「-n dbeng12/dbsrv12 サーバーオプション」257 ページ
- 「-xd dbeng12/dbsrv12 サーバーオプション」292 ページ
- 「接続パラメーターと接続文字列」30 ページ
- 「組み込みデータベースの接続」28 ページ


### 例

サーバー Guelph に接続します。

```
Server=Guelph
```

## StartLine (START) 接続パラメーター

アプリケーションからローカルデータベースサーバーを起動します。

### 構文

{ **StartLine** | **START** }=*local-database-server-command*

### 使用法

組み込みデータベース

### 指定可能な値

- **local-database-server-command**　デフォルトでは、SQL Anywhere は、実行中のデータベースサーバーに接続しようとします。指定された接続パラメーターでサーバーが見つからない場合、SQL Anywhere は *local-database-server-command* を使用して、新しいローカルデータベースサーバーを自動的に起動します。Host 接続パラメーターが指定されているか、CommLinks (LINKS) パラメーターに TCPIP が含まれている場合、データベースサーバーは自動的には起動しません。「SQL Anywhere データベースサーバーの構文」189 ページを参照してください。

### デフォルト

dbeng12 (Windows Mobile を除くすべてのプラットフォームの場合)

dbsrv12 (Windows Mobile の場合)

### 備考

StartLine 接続パラメーターは、指定のデータベースサーバーに接続できない場合にデータベースサーバーを起動するためにのみ使用されます。

たとえば、次のように、データベースを実行するデータベースサーバーを起動するとします。

```
dbeng12 c:¥mydb.db
```

別のデータベースに接続します (Server 接続パラメーターを使用したデータベースサーバー名の指定は行いません)。

```
dbisql -c "START=dbsrv12 -x none -c 8M;DBN=seconddb;DBF=c:¥myseconddb.db;UID=DBA;PWD=sql"
```

この場合、dbsrv12 データベースサーバーは起動しません。その代わりに、*mydb.db* の起動に使用された dbeng12 データベースサーバーが、*myseconddb.db* を起動して接続するために使用されます。

ただし、Server=*server-name* が指定されており、*server-name* という名前のデータベースサーバーが実行されていない場合は、dbsrv12 データベースサーバーが起動されます。

**注意**

データベース名、データベースファイル、サーバーを指定する場合は、StartLine 接続パラメーターではなく、DBN、DBF、Server 接続パラメーターを使用することをおすすめします。

次のコマンドでは、推奨される構文が使用されています。

```
START=dbeng12 -c 8M;Server=mydb;DBN=mydb;DBF=c:¥sample.db
```

次の構文はおすすめしません。

```
START=dbeng12 -c 8M -n mydb "c:¥sample.db"
```

### 参照


- 「StartLine [START] 接続パラメーターの使用」28 ページ
- 「接続パラメーターと接続文字列」30 ページ
- 「CommLinks (LINKS) 接続パラメーター」327 ページ
- 「組み込みデータベースの接続」28 ページ


### 例

次のデータソースフラグメントは、8 MB のキャッシュでパーソナルデータベースサーバーを起動します。

```
"StartLine=dbeng12 -c 8M;DBF=%SQLANYSAMP12%¥demo.db"
```

## Unconditional (UNC) 接続パラメーター

データベースサーバーへの接続があるときでも、db_stop_engine 関数を使用してデータベースサーバーを停止します。または、db_stop_database 関数を使用してデータベースを停止します。

### 構文

**{ Unconditional | UNC }={ YES | NO }**

### 使用法

db_stop_engine 関数と db_stop_database 関数のみ。

### デフォルト

NO

### 備考

db_stop_engine 関数は、データベースサーバーを停止します。db_stop_database 関数は、データベースを停止します。接続文字列で UNC=YES を指定すると、データベースサーバーまたはデータベースはアクティブな接続があるときでも停止されます。Unconditional が YES に設定されていない場合は、アクティブな接続がないときのみデータベースサーバーまたはデータベースが停止されます。

### 参照


- 「db_stop_database 関数」『SQL Anywhere サーバー プログラミング』
- 「db_stop_engine 関数」『SQL Anywhere サーバー プログラミング』
- 「接続パラメーターと接続文字列」30 ページ


## Userid (UID) 接続パラメーター

データベースへのログインに使用するユーザー ID を指定します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 構文

{ **Userid** | **UID** }=*userid*

### 使用法

特に制限なし

### 指定可能な値

- **userid**　ユーザー ID には、次に該当する値を指定できません。
  - 空白スペース、一重引用符、または二重引用符で始まる値
  - 空白スペースで終わる値
  - セミコロンを含む値

### デフォルト

なし

### 備考

データベースに接続するときは、統合化ログインまたは Kerberos ログインを使用している場合を除き、必ずユーザー ID を指定します。

### 参照


- 「Password (PWD) 接続パラメーター」360 ページ
- 「接続パラメーターと接続文字列」30 ページ
- 「ユーザー ID、権限、パーミッション」512 ページ


### 例

次の接続文字列フラグメントは、ユーザー ID DBA とパスワード sql を指定します。

```
UID=DBA;PWD=sql
```

# ネットワークプロトコルオプション

ネットワークプロトコルオプション (クライアントとサーバー両方のための) を使用して、さまざまなネットワークプロトコルの問題に対処できます。

| サーバー側 | クライアント側 |
| --- | --- |
| ネットワークプロトコルオプションは、サーバーコマンドに指定できます。次に例を示します。<br><br>dbsrv12 -x tcpip(PARM1=value1;PARM2=value2;...) | **注意**<br>ほとんどの場合、HOST 接続パラメーターを使用してください。 「Host 接続パラメーター」346 ページを参照してください。<br><br>CommLinks (LINKS) 接続パラメーターは、Host または ServerPort (PORT) 以外の TCP/IP オプションを指定する必要がある場合にのみ使用してください。接続文字列に CommLinks と Host の両方は指定できません。<br><br>クライアント側では、CommLinks (LINKS) 接続パラメーターとしてプロトコルオプションを入力できます。<br><br>CommLinks=tcpip(PARM1=value1;PARM2=value2;...)<br><br>**注意**<br>クライアントととして同じコンピューター上のデータベースサーバーに接続する場合は、共有メモリをおすすめします。「通信プロトコルオプションの考慮事項」97 ページを参照してください。 |

パラメーターにスペースがある場合、ネットワークプロトコルオプションを二重引用符で囲むことにより、システムコマンドインタプリタによって適切に解析されます。

```
dbsrv12 -x "tcpip(PARM1=value1;PARM2=value2;...)"
"CommLinks=tcpip(PARM1=value1;PARM2=value2;...)"
```

UNIX では、セミコロンがコマンドの区切り文字として解釈されるため、複数のパラメーターを指定する場合にも二重引用符が必要です。

ブールパラメーターは、YES、Y、ON、TRUE、T、1 のいずれかによってオンになり、NO、N、OFF、FALSE、F、0 のいずれかによってオフになります。パラメーターは、大文字と小文字を区別しません。

プロトコルオプションを設定ファイルに記述し、サーバーオプションの *@data* を使用して設定ファイルを呼び出すこともできます。

| TCP/IP | HTTP | HTTPS | TLS |
| --- | --- | --- | --- |
| 「Broadcast (BCAST) プロトコルオプション」 | 「DatabaseName (DBN) プロトコルオプション」 | 「certificate_company プロトコルオプション」 | 「certificate_company プロトコルオプション」 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| TCP/IP | HTTP | HTTPS | TLS |
|---|---|---|---|
| 「BroadcastListener (BLISTENER) プロトコルオプション (サーバー側のみ)」 | 「KeepaliveTimeout (KTO) プロトコルオプション」 | 「certificate_name プロトコルオプション」 | 「certificate_name プロトコルオプション」 |
| 「ClientPort (CPORT) プロトコルオプション (クライアント側のみ)」 | 「LocalOnly (LO) プロトコルオプション」 | 「certificate_unit プロトコルオプション」 | 「certificate_unit プロトコルオプション」 |
| 「DoBroadcast (DOBROAD) プロトコルオプション」 | 「LogFile (LOG) プロトコルオプション」 | 「DatabaseName (DBN) プロトコルオプション」 | 「FIPS プロトコルオプション」 |
| 「Host (IP) プロトコルオプション (クライアント側のみ)」 | 「LogFormat (LF) プロトコルオプション」 | 「Identity プロトコルオプション」 | 「LocalOnly (LO) プロトコルオプション」 |
| 「LDAP プロトコルオプション」 | 「LogMaxSize (LSIZE) プロトコルオプション」 | 「Identity_Password プロトコルオプション」 | 「trusted_certificate プロトコルオプション」 |
| 「LocalOnly (LO) プロトコルオプション」 | 「LogOptions (LOPT) プロトコルオプション」 | 「FIPS プロトコルオプション」 | |
| 「MyIP (ME) プロトコルオプション」 | 「MaxConnections (MAXCONN) プロトコルオプション」 | 「LocalOnly (LO) プロトコルオプション」 | |
| 「ReceiveBufferSize (RCVBUFSZ) プロトコルオプション」 | 「MaxRequestSize (MAXSIZE) プロトコルオプション」 | 「LogFile (LOG) プロトコルオプション」 | |
| 「SendBufferSize (SNDBUFSZ) プロトコルオプション」 | 「MyIP (ME) プロトコルオプション」 | 「LogFormat (LF) プロトコルオプション」 | |
| 「ServerPort (PORT) プロトコルオプション」 | 「ServerPort (PORT) プロトコルオプション」 | 「LogMaxSize (LSIZE) プロトコルオプション」 | |
| 「TDS プロトコルオプション (サーバー側のみ)」 | 「Timeout (TO) プロトコルオプション」 | 「LogOptions (LOPT) プロトコルオプション」 | |

| TCP/IP | HTTP | HTTPS | TLS |
|---|---|---|---|
| 「Timeout (TO) プロトコルオプション」 | | 「MaxConnections (MAXCONN) プロトコルオプション」 | |
| 「VerifyServerName (VERIFY) プロトコルオプション (クライアント側のみ)」 | | 「MaxRequestSize (MAXSIZE) プロトコルオプション」 | |
| | | 「MyIP (ME) プロトコルオプション」 | |
| | | 「ServerPort (PORT) プロトコルオプション」 | |
| | | 「Timeout (TO) プロトコルオプション」 | |
| | | 「trusted_certificate プロトコルオプション」 | |

# Broadcast (BCAST) プロトコルオプション

サーバーの検索に使用する IP ブロードキャストアドレスを指定します。

### 構文

{ **Broadcast** | **BCAST** }=*ip-address*

### 使用可能なプロトコル

TCP/IP

### 指定可能な値

- **ip-address**　この文字列は IP アドレス形式で指定する必要があります。

### デフォルト

同一サブネット上のすべてのアドレスにブロードキャストします。

### 備考

デフォルトのブロードキャストアドレスは、ローカル IP アドレスとサブネットマスクを使用して作成されます。サブネットマスクは、IP アドレスのどの部分がネットワークを指定し、どの部分がホストを指定するかを示します。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

たとえば、マスクが 255.255.255.0 のサブネット 10.24.98.x の場合、デフォルトのブロードキャストアドレスは 10.24.98.255 になります。

Windows プラットフォームで IPv6 アドレスを指定する場合、インターフェイス識別子を使用する必要があります。UNIX プラットフォームでは、IPv6 アドレスのインターフェイス識別子とインターフェイス名の両方がサポートされます。Linux (カーネル 2.6.13 以降) では、インターフェイス識別子が必要です。

### 参照


- 「SQL Anywhere での IPv6 サポート」100 ページ
- 「BroadcastListener (BLISTENER) プロトコルオプション (サーバー側のみ)」373 ページ
- 「DoBroadcast (DOBROAD) プロトコルオプション」379 ページ
- 「Broadcast Repeater ユーティリティを使用したデータベースサーバーの検出」82 ページ


### 例

次の接続文字列の例は、IPv6 を使用する場合にインターフェイス番号 2 だけでブロードキャストするようクライアントに指示します。

```
LINKS=tcpip(BROADCAST=ff02::1%2)
```

次の接続文字列の例は、クライアントに対して、IPv4 を使用するときに 10.24.98.0 から 10.24.98.254 までの範囲ですべての IP アドレスにブロードキャストするように通知します。

```
LINKS=tcpip(BROADCAST=10.24.98.255)
```

## BroadcastListener (BLISTENER) プロトコルオプション (サーバー側のみ)

指定したポートのブロードキャスト受信を制御します。

### 構文

**{ BroadcastListener | BLISTENER }={ YES | NO }**

### 使用可能なプロトコル

TCP/IP (サーバー側のみ)

### デフォルト

YES

### 備考

このオプションを使用して、このポートのブロードキャスト受信をオフに設定できます。

**-sb 0** を指定することは、TCP/IP で BroadcastListener=NO に設定するのと同じことです。

ブロードキャスト受信が OFF の場合、データベースサーバーが UDP のブロードキャストに応答しません。したがって、クライアントは HOST プロトコルオプションを使用してデータベース

サーバーのホスト名を指定するか、データベースサーバーを LDAP に登録し、クライアントで LDAP を使用してデータベースサーバーを検索する必要があります。また、dblocate ユーティリティの出力にはデータベースサーバーが含まれません。

### 参照


- 「-sb dbeng12/dbsrv12 サーバーオプション」270 ページ
- 「Broadcast (BCAST) プロトコルオプション」372 ページ
- 「DoBroadcast (DOBROAD) プロトコルオプション」379 ページ
- 「サーバー列挙ユーティリティ (dblocate)」999 ページ
- 「Host 接続パラメーター」346 ページ
- 「Host (IP) プロトコルオプション (クライアント側のみ)」381 ページ


### 例

TCP/IP 接続に対応するサーバーのうち、Host 接続パラメーターまたは Host プロトコルオプションを使用した TCP/IP 接続を必要とするデータベースサーバーを起動します。

```
dbsrv12 -x tcpip(BroadcastListener=NO) ...
```

次に示すのは、データベースサーバーに接続するための接続文字列フラグメントです。

```
...LINKS=tcpip(...HOST=myhost;...);...
```

## certificate_company プロトコルオプション

証明書に記されている組織フィールドがこの値と一致する場合にだけ、クライアントでサーバー証明書を受け入れるようにします。

### 構文

**certificate_company=** *organization*

### 使用可能なプロトコル

TLS、HTTPS

### デフォルト

なし

### 備考

SQL Anywhere クライアントは認証局が署名した証明書をすべて信頼するため、同じ認証局が他の会社用に発行した証明書も信頼してしまうことがあります。識別方法がないままだと、クライアントは競争相手のデータベースサーバーを自分の会社のものだと勘違いし、誤って機密性の高い情報を送信してしまう可能性があります。このオプションによって追加の検証が指定され、証明書の識別情報部分にある組織フィールドが、指定した特定の値と照合されます。

HTTPS は、Web サービスのクライアントプロシージャーだけでサポートされています。「CREATE PROCEDURE 文 [Web サービス]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**注意**
別途ライセンスが必要な必須コンポーネント

ECC 暗号化と FIPS 認定の暗号化には、別途ライセンスが必要です。強力な暗号化テクノロジはすべて、輸出規制対象品目です。

「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

### 参照


- 「certificate_name プロトコルオプション」375 ページ
- 「certificate_unit プロトコルオプション」376 ページ
- 「trusted_certificate プロトコルオプション」401 ページ
- 「Encryption (ENC) 接続パラメーター」342 ページ
- 「SQL Anywhere クライアント／サーバー通信の暗号化」1420 ページ
- 「証明書作成ユーティリティ (createcert)」924 ページ
- 「-xs dbeng12/dbsrv12 サーバーオプション」294 ページ


### 例

次のコマンドは、トランスポートレイヤーセキュリティを使用して SQL Anywhere のサンプルデータベースを Interactive SQL に接続します。

```
dbisql -c
"UID=DBA;PWD=sql;Host=myhost;Server=demo;ENC=TLS(
tls_type=RSA;FIPS=n;trusted_certificate=c:¥temp¥myident;
certificate_unit='SA';certificate_company='Sybase iAnywhere';
certificate_name='Sybase')"
```

## certificate_name プロトコルオプション

証明書に記されている通称フィールドがこの値と一致する場合にだけ、クライアントでサーバー証明書を受け入れるようにします。

### 構文

**certificate_name=***common-name*

### 使用可能なプロトコル

TLS、HTTPS

### デフォルト

なし

### 戻り値

SQL Anywhere クライアントは認証局が署名した証明書をすべて信頼するため、同じ認証局が他の会社用に発行した証明書も信頼してしまうことがあります。識別方法がないままだと、クライアントは競争相手のデータベースサーバーを自分の会社のものだと勘違いし、誤って機密性の高

い情報を送信してしまう可能性があります。このオプションによって追加の検証が指定され、証明書の識別情報部分にある通称フィールドが、指定した特定の値と照合されます。

HTTPS は、Web サービスのクライアントプロシージャーだけでサポートされています。「CREATE PROCEDURE 文 [Web サービス]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

**注意**
別途ライセンスが必要な必須コンポーネント

ECC 暗号化と FIPS 認定の暗号化には、別途ライセンスが必要です。強力な暗号化テクノロジはすべて、輸出規制対象品目です。

「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

### 参照

- 「certificate_company プロトコルオプション」374 ページ
- 「certificate_unit プロトコルオプション」376 ページ
- 「trusted_certificate プロトコルオプション」401 ページ
- 「Encryption (ENC) 接続パラメーター」342 ページ
- 「SQL Anywhere クライアント／サーバー通信の暗号化」1420 ページ
- 「証明書作成ユーティリティ (createcert)」924 ページ
- 「-xs dbeng12/dbsrv12 サーバーオプション」294 ページ

### 例

次のコマンドは、トランスポートレイヤーセキュリティを使用して SQL Anywhere のサンプルデータベースを Interactive SQL に接続します。

```
dbisql -c
"UID=DBA;PWD=sql;HOST=myhost;Server=demo;ENC=TLS(
tls_type=RSA;FIPS=n;trusted_certificate=c:¥temp¥myident;
certificate_unit='SA';certificate_company='Sybase iAnywhere';
certificate_name='Sybase')"
```

## certificate_unit プロトコルオプション

証明書に記されている組織単位フィールドがこの値と一致する場合にだけ、クライアントでサーバー証明書を受け入れるようにします。

### 構文

**certificate_unit=***organization-unit*

### 使用可能なプロトコル

TLS、HTTPS

### デフォルト

なし

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 備考

SQL Anywhere クライアントは認証局が署名した証明書をすべて信頼するため、同じ認証局が他の会社用に発行した証明書も信頼してしまうことがあります。識別方法がないままだと、クライアントは競争相手のデータベースサーバーを自分の会社のものだと勘違いし、誤って機密性の高い情報を送信してしまう可能性があります。このオプションによって追加の検証が指定され、証明書の識別情報部分にある組織単位フィールドが、指定した特定の値と照合されます。

HTTPS は、Web サービスのクライアントプロシージャーだけでサポートされています。「CREATE PROCEDURE 文 [Web サービス]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

**注意**
別途ライセンスが必要な必須コンポーネント

ECC 暗号化と FIPS 認定の暗号化には、別途ライセンスが必要です。強力な暗号化テクノロジはすべて、輸出規制対象品目です。

「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

### 参照


- 「certificate_company プロトコルオプション」374 ページ
- 「certificate_name プロトコルオプション」375 ページ
- 「trusted_certificate プロトコルオプション」401 ページ
- 「Encryption (ENC) 接続パラメーター」342 ページ
- 「SQL Anywhere クライアント／サーバー通信の暗号化」1420 ページ
- 「証明書作成ユーティリティ (createcert)」924 ページ
- 「-xs dbeng12/dbsrv12 サーバーオプション」294 ページ


### 例

次のコマンドは、トランスポートレイヤーセキュリティを使用して SQL Anywhere のサンプルデータベースを Interactive SQL に接続します。

```
dbisql -c
"UID=DBA;PWD=sql;Host=myhost;Server=demo;ENC=TLS(
tls_type=RSA;FIPS=n;trusted_certificate=c:¥temp¥myident;
certificate_unit='SA';certificate_company='Sybase iAnywhere';
certificate_name='Sybase')"
```

## ClientPort (CPORT) プロトコルオプション (クライアント側のみ)

TCP/IP を使用してクライアントアプリケーションが通信するポート番号を指定します。

### 構文

{ **ClientPort** | **CPORT** }=*port-number*

### 使用可能なプロトコル

TCP/IP (クライアント側のみ)

### デフォルト

ネットワークの実装によって、接続ごとに動的に割り当てられます。ファイアウォールの制限がない場合は、このプロトコルオプションを使用しないようおすすめします。

### 備考

このオプションは、ファイアウォールを介した接続のために提供されています。ファイアウォールソフトウェアは、TCP/UDP ポートに従ってフィルターします。ファイアウォールの理由によって必要な場合以外は、このプロトコルオプションを使用しないようおすすめします。

ClientPort オプションは、クライアントアプリケーションが TCP/IP を使って通信するポート番号を指定します。単一のポート番号、または個々のポート番号の組み合わせやポート番号の範囲を指定できます。次に例を示します。

- (CPORT=1234)
- (CPORT=1234,1235,1239)
- (CPORT=1234-1238)
- (CPORT=1234-1237,1239,1242)

指定されたデータソースや接続文字列を使用して複数の接続を確立する場合、ポート番号のリストや範囲を指定することをおすすめします。ポート番号を 1 つだけ指定すると、アプリケーションが維持できるのは、一度に 1 つの接続のみとなります。また、1 つの接続を閉じた後は、数分のタイムアウト時間が生じます。その間、指定されたポートを使って新しい接続は作成できません。ポート番号のリストや範囲を指定すると、アプリケーションは、いずれかのポート番号との接続が確立するまで、試行を続けます。

### 参照


- 「Host (IP) プロトコルオプション (クライアント側のみ)」381 ページ
- 「DoBroadcast (DOBROAD) プロトコルオプション」379 ページ
- 「ServerPort (PORT) プロトコルオプション」396 ページ
- 「通信プロトコル」96 ページ
- 「ファイアウォール経由の接続」101 ページ


### 例

次の接続文字列フラグメントは、ポート 6000 を使用するアプリケーションから、ポート 5000 を使用する my-server という名前のサーバーへの接続を確立します。

```
CommLinks=tcpip(ClientPort=6000;ServerPort=5000);ServerName=my-server
```

次の接続文字列フラグメントは、指定されたポートを使用できるアプリケーションから、デフォルトのサーバーポートを使用する my-server という名前のサーバーに通信する接続を確立します。

```
CommLinks=tcpip(ClientPort=5040,5050-5060,5070);
ServerName=my-server
```

# DatabaseName (DBN) プロトコルオプション

Web 要求を処理するときに使用するデータベースの名前を指定します。また、REQUIRED や AUTO キーワードを使用して URL の一部としてデータベース名が必要かどうかを指定します。

### 構文

{ **DatabaseName** | **DBN** }={ **AUTO** | **REQUIRED** | *database-name* }

### 使用可能なプロトコル

HTTP、HTTPS

### デフォルト

AUTO

### 戻り値

このプロトコルオプションが REQUIRED に設定されている場合は、URL がデータベース名を指定します。

このプロトコルオプションが AUTO に設定されている場合は、URL がデータベース名を指定できますが、必須ではありません。URL にデータベース名が含まれていない場合は、サーバーでのデフォルトのデータベースを Web 要求の処理に使用します。AUTO に設定されている場合、サーバーは URL にデータベース名が含まれているかどうかを判別しなければならないため、あいまいにならないように Web サイトを設計してください。

このプロトコルオプションにデータベースが設定されている場合は、このデータベースを使用してすべての Web 要求を処理します。URL にはデータベース名を含めないでください。

### 例

次のコマンドは 2 つのデータベースを起動しますが、HTTP 経由でのアクセスを許可されているのはそのうちの 1 つだけです。

```
dbsrv12 -xs http(DBN=web) "%SQLANY12%¥demo.db" web.db
```

次のコマンドは、2 つの HTTP Web サービスを起動します。1 つは *your-first-database.db* 用、もう 1 つは *your-second-database.db* 用です。

```
dbsrv12 -xs http(port=80;dbn=your-first-database),http(port=8800;dbn=your-second-database) your-first-database.db your-second-database.db
```

# DoBroadcast (DOBROAD) プロトコルオプション

クライアントがデータベースサーバーを検索する方法と、データベースサーバーが起動時にブロードキャストを実行するかどうかを制御します。

### 構文 (データベースサーバー)

{ **DoBroadcast** | **DOBROAD** }= { **YES** | **NO** }

### 構文 (クライアント)

{ **DoBroadcast** | **DOBROAD** }= { **ALL** | **NONE** | **DIRECT** }

### 使用可能なプロトコル

TCP/IP

### 指定可能な値 (データベースサーバー)

- **YES**　DoBroadcast=YES に設定すると、データベースサーバーは起動時にブロードキャストを実行して、同じ名前の他のデータベースサーバーを検索できます。

- **NO**　DoBroadcast=NO に設定すると、起動時にデータベースサーバーがブロードキャストを実行して、同じ名前の他のデータベースサーバーを検索しないようにできます。この設定が役に立つ場合もまれにありますが、通常は必要ありません。

### 指定可能な値 (クライアント)

- **ALL**　DoBroadcast=ALL に設定した場合、ブロードキャストを実行してデータベースサーバーを検索します。最初は、ローカルサブネットにブロードキャストされます。HOST を指定した場合、各ホストにはブロードキャストパケットも送信されます。ブロードキャストパケットはすべて UDP パケットです。

- **NONE**　DoBroadcast=NONE を指定すると、UDP ブロードキャストは使用されず、データベースサーバーアドレスキャッシュ (*sasrv.ini* ファイルに存在) は無視されます。指定した HOST/PORT との TCP/IP 接続が直接行われ、サーバー名が検証されます。DoBroadcast=DIRECT を指定する場合は、HOST (IP) プロトコルオプションが必要です。

- **DIRECT**　DoBroadcast=DIRECT を設定した場合、データベースサーバーを検索するときに、ローカルサブネットへのブロードキャストは実行されません。ブロードキャストパケットは、HOST (IP) プロトコルオプションリストにあるホストにのみ送信されます。DoBroadcast=DIRECT を指定する場合は、HOST (IP) プロトコルオプションが必要です。

### デフォルト

YES (データベースサーバー)

ALL (クライアント) HOST (IP) プロトコルオプションが指定された場合、デフォルトは DIRECT です。

### 戻り値

**クライアントでの使用法**　TCP/IP の場合は、VerifyServerName (VERIFY) プロトコルオプションを NO に設定して、サーバー名を検証しないようにすることもできます。HOST (IP) プロトコルオプションは LDAP を使用していない場合は必須のパラメーターですが、ServerPort (PORT) プロトコルオプションは省略可能です。

DIRECT と NONE の場合は、HOST オプションでデータベースサーバーホストを指定します。

### 参照


- 「Broadcast (BCAST) プロトコルオプション」372 ページ
- 「BroadcastListener (BLISTENER) プロトコルオプション (サーバー側のみ)」373 ページ


### 例

次の接続文字列は、silver という名前のコンピューター上で実行している gold という名前のデータベースサーバーに接続します。

```
Host=silver;ServerName=gold
```

次の接続文字列は、CommLinks プロトコルオプションを使用して同じデータベースサーバーに接続します。

```
CommLinks=tcpip(DOBROADCAST=DIRECT;HOST=silver);ServerName=gold
```

## Host (IP) プロトコルオプション (クライアント側のみ)

データベースサーバーのアドレスを指定します。

### 構文

{ **Host** | **IP** }=*ip-address*

### 使用可能なプロトコル

TCP/IP (クライアント側のみ)

### 指定可能な値

- **ip-address**　この文字列は、IP アドレス形式で指定する必要があり、任意でポート番号を含めることができます (コロンで区切る)。ホスト値のリストは、カンマで区切って指定します。IPv6 アドレスの場合は、アドレスを角カッコまたはカッコで囲む必要があります。**localhost** を使用して現在のコンピューターを識別できます。次に例を示します。

```
links=tcpip(HOST=myhost)
links=tcpip(HOST=myhost:1234)
links=tcpip(HOST=10.25.13.5,myotherhost)
links=tcpip(HOST=myhost:1234,10.25.65.112)
links=tcpip(HOST=myhost:1234,myotherhost:5678)
```

### デフォルト

現在の TCP/IP サブセットを検索します。

### 備考

**注意**
HOST プロトコルオプションは、Host 接続パラメーターとは別のものです。HOST プロトコルオプションは、CommLinks 接続パラメーターによって使用されます。CommLinks (LINKS) 接続パラメーターは、Host または ServerPort (PORT) 以外の TCP/IP オプションを指定する必要がある場合にのみ使用してください。接続文字列に CommLinks と Host の両方は指定できません。
「CommLinks (LINKS) 接続パラメーター」327 ページを参照してください。

ほとんどの場合、HOST 接続パラメーターを使用してください。「Host 接続パラメーター」346 ページを参照してください。

HOST は、データベースサーバーのアドレスを指定します。IP と HOST は同義語です。カンマで区切ったアドレスのリストを使って、複数のコンピューターを検索できます。また、コロンを区切り文字として使用してポート番号を IP アドレスに追加できます。別の方法として、HOST=myhost;PORT=5000 のように、PORT プロトコルオプションを明示的に使用してホストとサーバーのポートを指定することもできます。IPv6 アドレスの場合は、アドレスを角カッコまたはカッコで囲む必要があります。例：[fe80::5445:5245:444f]:2638。ポートを指定した場合は、そのポート番号のみが TCP/IP 接続と UDP ブロードキャストに使用されます。ポート番号を指定しない場合は、ポート 2638 が使用されます。

Windows プラットフォームで IPv6 アドレスを指定する場合、インターフェイス識別子を使用する必要があります。UNIX プラットフォームでは、IPv6 アドレスのインターフェイス識別子とインターフェイス名の両方がサポートされます。Linux (カーネル 2.6.13 以降) では、インターフェイス識別子が必要です。

-z オプションを使用すると、データベースサーバーに、起動時にデータベースサーバーメッセージウィンドウでアドレス情報を表示させることができます。

LogFile 接続パラメーターを指定すると、クライアントアプリケーションに、アドレス指定情報をログファイルに書き込ませることができます。

HOST プロトコルオプションまたは HOST 接続パラメーターと共に LocalOnly プロトコルオプションを指定することはできません。

### 参照


- 「SQL Anywhere での IPv6 サポート」100 ページ
- 「-z dbeng12/dbsrv12 サーバーオプション」296 ページ
- 「LogFile (LOG) 接続パラメーター」354 ページ
- 「ClientPort (CPORT) プロトコルオプション (クライアント側のみ)」377 ページ


### 例

次の接続文字列フラグメントは、kangaroo と 197.75.209.222 (ポート 2369) というコンピューターを検索し、データベースサーバーを見つけるようにクライアントに指示します。

```
LINKS=tcpip(IP=kangaroo,197.75.209.222:2369)
```

次の接続文字列フラグメントは、my-server と kangaroo というコンピューターを検索し、データベースサーバーを見つけるようにクライアントに指示します。

```
LINKS=tcpip(HOST=my-server,kangaroo;PORT=2639)
```

次の接続文字列フラグメントは、ポート 1234 で稼働する host1 上のサーバーと、ポート 4567 で稼働する host2 上のサーバーを検索するようにクライアントに指示します。クライアントは、ポート 4567 の host1 またはポート 1234 の host2 は検索しません。

```
LINKS=tcpip(HOST=host1:1234,host2:4567)
```

次の接続文字列フラグメントは、IPv6 アドレス上でサーバーを探すようクライアントに指示します。

```
LINKS=tcpip(HOST=fe80::5445:5245:444f)
```

次の接続文字列フラグメントは、ポート 2639 の IPv6 アドレス上でデータベースサーバーを探すようクライアントに指示します。

```
LINKS=tcpip(HOST=[fe80::5445:5245:444f]:2639)
```

Host プロトコルオプションを指定した IPv6 アドレスの使用例を次に示します。

```
// Global scope address, unique everywhere, so no interface index is required
// no index required
-c "links=tcpip(Host=fd77:55d:59d9:56a:202:55ff:fe76:df19)"
// all communication is done through interface 2
-c "links=tcpip(Host=fd77:55d:59d9:56a:202:55ff:fe76:df19%2)"
// all communication is done through eth0
-c "links=tcpip(Host=fd77:55d:59d9:56a:202:55ff:fe76:df19%eth0)"

// Link scope address, addresses are unique on each interface
// possibly ambiguous (this host may exist through both eth0 and eth1)
-c "links=tcpip(Host=fe80::202:55ff:fe76:df19)"
// not ambiguous because it must use interface 2
-c "links=tcpip(Host=fe80::202:55ff:fe76:df19%2)"
// not ambiguous because it must use eth0
-c "links=tcpip(Host=fe80::202:55ff:fe76:df19%eth0)"
```

# Identity プロトコルオプション

ID ファイルの名前を指定します。

### 構文

**Identity=***identity-file*

### 使用可能なプロトコル

HTTPS

### 指定可能な値

- **identity-file**　この文字列は、ID ファイルの名前を指定します。

### デフォルト

デフォルトの ID ファイル名はありません。

### 備考

トランスポートレイヤーセキュリティを使用している場合、ID ファイルには、パブリック証明書とプライベートキーが含まれており、自己署名されない証明書の場合は、さらに署名を行うすべての証明書も含まれています。これには暗号化証明書も含まれます。この証明書のパスワードは、Identity_Password パラメーターで指定します。

### 参照


- 「トランスポートレイヤーセキュリティの設定」1412 ページ
- 「Identity_Password プロトコルオプション」384 ページ


### 例

特定の暗号化された証明書を使用するために、Web 接続が必要なサーバーを起動します。

```
dbsrv12 -xs https(Identity=cert.file;Identity_Password=secret) ...
```

> **警告**
> サンプル ID ファイルは、テスト作業と開発作業にのみ使用します。この証明書は SQL Anywhere の標準部分であるため、保護機能は備えていません。アプリケーションを配備する前に独自の証明書で置換してください。

## Identity_Password プロトコルオプション

暗号化証明書のパスワードを指定します。

### 構文

**Identity_Password=***password*

### 使用可能なプロトコル

HTTPS

### 指定可能な値

- **password**　この文字列は、暗号化証明書のパスワードを指定します。

### デフォルト

デフォルトの ID ファイルのパスワードはありません。

### 備考

トランスポートレイヤーセキュリティを使用している場合、このオプションにより、Identity プロトコルオプションで指定した暗号化証明書に対応するパスワードを指定します。

### 参照


- 「トランスポートレイヤーセキュリティの設定」1412 ページ
- 「Identity プロトコルオプション」383 ページ



Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 例

特定の暗号化された証明書を使用するために、Web 接続が必要なサーバーを起動します。

```
dbsrv12 -xs https(Identity=cert.file;Identity_Password=secret) ...
```

## FIPS プロトコルオプション

FIPS 認定のセキュリティアルゴリズムを使用して、データベースファイルを暗号化したり、データベースクライアント／サーバー通信や Web サービスを暗号化できます。

> **注意**
> 別途ライセンスが必要な必須コンポーネント
>
> ECC 暗号化と FIPS 認定の暗号化には、別途ライセンスが必要です。強力な暗号化テクノロジはすべて、輸出規制対象品目です。
>
> 「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

### 構文

**{ FIPS }={ YES | NO }**

### 使用可能なプロトコル

TLS、HTTPS

### 指定可能な値

YES、NO

### デフォルト

なし

### 備考

FIPS 認定の暗号は、RSA 暗号化でのみでサポートされています。有効な fips プロトコルオプションを持たないクライアントは、fips オプションが有効なサーバーに接続できます。逆についても同様です。このオプションは、エンドツーエンド暗号化と組み合わせて使用できます。

fips が YES に設定されている場合、Mobile Link クライアントは、FIPS 140-2 基準を満たした RSA 実装および AES 実装を使用します。このオプションは、ECC の使用時にはサポートされません。

### 参照


- 「certificate_company プロトコルオプション」374 ページ
- 「certificate_name プロトコルオプション」375 ページ
- 「certificate_unit プロトコルオプション」376 ページ
- 「trusted_certificate プロトコルオプション」401 ページ
- 「Encryption (ENC) 接続パラメーター」342 ページ
- 「SQL Anywhere クライアント／サーバー通信の暗号化」1420 ページ
- 「証明書作成ユーティリティ (createcert)」924 ページ
- 「-ec dbeng12/dbsrv12 サーバーオプション」223 ページ
- 「-xs dbeng12/dbsrv12 サーバーオプション」294 ページ


### 例

次のコマンドラインは、Web サーバーの PORT、FIPS、Identity、Identity_Password ネットワークプロトコルオプションを設定します。

```
dbsrv12 -xs https(PORT=544;FIPS=YES;
   Identity=certificate.id;Identity_Password=password) your-database-name.db
```

## KeepaliveTimeout (KTO) プロトコルオプション

HTTP 要求間の最大アイドル時間を秒単位で指定します。

### 構文

{ **KeepaliveTimeout** | **KTO** }=*timeout-value*

### 使用可能なプロトコル

HTTP

### 指定可能な値

- **timeout-value**　この整数は、HTTP 要求間に待機する時間数を秒単位で指定します。接続をタイムアウトしないようにする場合は、KTO=0 を指定します。運用環境では、悪質なクライアントがデータベースサーバーのリソースを消費して、他のクライアントが接続できなくなる可能性があるので、KTO=0 に設定することはおすすめしません。

### デフォルト

60

### 戻り値

このプロトコルオプションは、キープアライブ HTTP 要求にのみ適用されます。これは、最新の HTTP 要求が受信された時間から、データが次の HTTP 要求用に最初に受信されるときまでに許可される最大アイドル時間を指定します。新しい要求用のデータが受信されると、Timeout プロトコルオプションで定義されたタイムアウト期間の計測が開始されます。

### 参照


- 「Timeout (TO) プロトコルオプション」400 ページ



Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

# LDAP プロトコルオプション

クライアントは、IP アドレスを指定せずにデータベースサーバーを検索できます。

### 構文

**LDAP={ YES | NO |** *filename* **}**

### 使用法

TCP/IP

### 指定可能な値

- **YES**　LDAP=YES と指定すると、LDAP のサポートがオンになり、*saldap.ini* (デフォルトファイル名) が設定ファイルとして使用されます。

- **NO**　NO を指定すると、LDAP サポートがオフになります。

- **filename**　LDAP=*filename* と指定すると、LDAP のサポートがオンになり、指定したファイルが設定ファイルとして使用されます。

### デフォルト

YES

### 備考

データベースサーバー自体を LDAP サーバーに登録することにより、クライアントが LDAP サーバーにクエリを実行できます。これにより、WAN 上で動作するクライアントやファイアウォールを経由するクライアントが IP アドレスを指定せずにサーバーを検索できます。また、検出ユーティリティ (dblocate) もそのようなサーバーを検索できるようになります。

ファイル難読化ユーティリティを使用すると、単純暗号化によって *saldap.ini* ファイルの内容を隠すことができます。「.ini ファイルの内容の非表示」465 ページを参照してください。

LDAP は TCP/IP でのみ使用されます。

### 参照

- 「LDAP サーバーを使用した接続」103 ページ

# LocalOnly (LO) プロトコルオプション

クライアントがローカルコンピューター上のサーバー (存在する場合) のみに接続できるようにします。

### 構文

**{ LocalOnly | LO }={ YES | NO }**

### 使用可能なプロトコル

TCP/IP、HTTP、HTTPS

### デフォルト

なし

### 戻り値

サーバー名が一致するデータベースサーバーが、ローカルコンピューターで見つからない場合、データベースサーバーは自動的に起動しません。

LocalOnly (LO) プロトコルオプションは、DoBroadcast=ALL (デフォルト) の場合にのみ役立ちます。

LocalOnly=YES を設定すると、サーバーから他のコンピューターへのブロードキャスト応答が無視された場合を除き、通常のブロードキャストメカニズムが使用されます。

LocalOnly (LO) プロトコルオプションをサーバーで使用して、ローカルコンピューターへの接続に限定させることができます。リモートコンピューターからの接続試行ではこのサーバーは検索されず、検出 (dblocate) ユーティリティからはこのサーバーは見えません。LocalOnly (LO) プロトコルオプションを YES に設定してサーバーを稼働すると、接続や CPU の制限を受けずにネットワークサーバーがパーソナルサーバーとして稼働します。

このパラメーターを YES に設定すると、ネットワークデータベースサーバーは異なるコンピューターで実行中のクライアントからの通信をすべて拒否します。このオプションは、他のコンピューターからの Web サービス要求を受け入れることのないパーソナルデータベースサーバーには影響しません。デフォルト値は NO で、どの場所にあるクライアントの要求も受け入れます。

HOST プロトコルオプションまたは HOST 接続パラメーターと共に LocalOnly プロトコルオプションを指定することはできません。

### 参照


- 「Broadcast (BCAST) プロトコルオプション」372 ページ
- 「HTTP Web サーバーの起動」『SQL Anywhere サーバー プログラミング』


## LogFile (LOG) プロトコルオプション

データベースサーバーが Web 要求に関する情報を書き込むファイル名を指定します。

### 構文

{ **LogFile** | **LOG** }=*filename*

### 使用可能なプロトコル

HTTP、HTTPS

### デフォルト

なし

### 戻り値

このプロトコルオプションは、データベースサーバーが Web 要求に関する情報を書き込むファイル名を指定します。

### 参照


- 「LogFormat (LF) プロトコルオプション」389 ページ
- 「LogMaxSize (LSIZE) プロトコルオプション」390 ページ
- 「LogOptions (LOPT) プロトコルオプション」391 ページ


## LogFormat (LF) プロトコルオプション

データベースサーバーが Web 要求に関する情報を書き込むログファイルに書き込まれるメッセージのフォーマットと、メッセージに表示されるフィールドを制御します。

### 構文

{ **LogFormat** | **LF** }=*format-string*

### 使用可能なプロトコル

HTTP、HTTPS

### 指定可能な値

- **format-string**　次のコードがサポートされます。
  - **@@**　@ 文字
  - **@B**　要求の処理が開始された日付／時刻 (エラーにより要求をキューイングできない場合を除く)
  - **@C**　クライアントが接続した日付／時刻
  - **@D**　要求に関連するデータベース名
  - **@E**　エラーが発生した場合の、エラーメッセージテキスト
  - **@F**　要求の処理が終了した日付／時刻
  - **@I**　クライアントの IP アドレス
  - **@J**　@I オプションで指定されたクライアントポートのログを取ります。
  - **@L**　ヘッダーと本文を含んだ応答の長さ (バイト)
  - **@M**　HTTP 要求方式

- **@P** 要求に関連するリスナーポート
- **@Q** 要求の処理がキューイングされた日付／時刻 (エラーにより要求をキューイングできない場合を除く)
- **@R** HTTP 応答のステータスコードおよび説明
- **@S** HTTP ステータスコード
- **@T** 現在のログエントリが書き込まれた日付／時刻
- **@U** 要求 URI
- **@V** 要求 HTTP バージョン
- **@W** 要求を処理した時間 (@F - @B)、またはエラーにより要求が処理されなかった場合は 0.000

### デフォルト

**@T - @W - @I:@J - @P - "@M @U @V" - @R - @L - @E**

### 備考

このプロトコルオプションは、Web 要求に関する情報を格納するログファイルに書き込まれるメッセージのフォーマットと、表示されるフィールドを制御します。文字列に表示される場合、各メッセージが書き込まれると現在の値がコードに置き換えられます。

HTTP 要求メソッドがサポートされていない、または URI のフォーマットが不正であるか必要なデータベース名が URI ないために Web 要求が失敗した場合は、HTTP メソッド (@M) と HTTP バージョン (@V) が文字列 **???** を返し、URI (@U) が **>>>** に続いて指定された要求を返します。

たとえば、ログフォーマットが「@M @U @V」に設定されている場合は、URL request/sample/test HTTP/1.0 の未知の HTTP メソッドが次の文字列を返します。

```
"???>>>request/sample/test HTTP/1.0???"
```

### 参照


- 「LogFile (LOG) プロトコルオプション」388 ページ
- 「LogMaxSize (LSIZE) プロトコルオプション」390 ページ
- 「LogOptions (LOPT) プロトコルオプション」391 ページ


## LogMaxSize (LSIZE) プロトコルオプション

データベースサーバーが Web 要求に関する情報を書き込むログファイルの最大サイズを制御します。

### 構文

{ **LogMaxSize** | **LSIZE** }=*size*[ **k** | **m** | **g** ]

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 使用可能なプロトコル

HTTP、HTTPS

### 指定可能な値

- **size**　この整数は、Web 要求情報が書き込まれるファイルの最大サイズを指定します。デフォルト値はバイト単位ですが、**k**、**m**、**g** のいずれかを使用してキロバイト、メガバイト、またはギガバイトの単位を指定できます。LogMaxSize が 0 の場合、ログファイルのサイズは無制限です。

### デフォルト

0

### 備考

ログファイルが指定したサイズに達すると、名前が変更されて、別のログファイルが作成されます。

### 参照


- 「LogFile (LOG) プロトコルオプション」388 ページ
- 「LogFormat (LF) プロトコルオプション」389 ページ
- 「LogOptions (LOPT) プロトコルオプション」391 ページ


## LogOptions (LOPT) プロトコルオプション

データベースサーバーが Web 要求に関する情報を書き込むログに記録されるメッセージタイプを指定します。

### 構文

**{ LogOptions | LOPT }= [ NONE ] [, OK ] [, INFO ] [, ERRORS ] [, ALL ] [,** *status-codes* **] [, REQHDRS ] [, RESHDRS ] [, HEADERS ]**

### 使用可能なプロトコル

HTTP、HTTPS

### 指定可能な値

次のキーワードは、ログ出力するメッセージのカテゴリを制御します。

- **NONE**　何もログ出力しない
- **OK**　ログ要求が正しく完了 (20x HTTP ステータスコード)
- **INFO**　終了または未変更ステータスコードを返すログ要求 (3xx HTTP ステータスコード)
- **ERRORS**　すべてのエラーのログを取る (4xx と 5xx HTTP ステータスコード)
- **USER**　すべてのエラーのログを取る (6xx、7xx、8xx、9xx HTTP ステータスコード)

- **ALL**　すべての要求をログ出力する

次の共通 HTTP ステータスコードも使用可能です。特定のステータスコードを返す要求をログ出力するために使用できます。

- **C200**　OK
- **C400**　不正な要求
- **C401**　無認可
- **C403**　禁止
- **C404**　見つからない
- **C408**　要求タイムアウト
- **C501**　未実装
- **C503**　サービス利用不可

次のキーワードを使用してログ出力されたメッセージの詳細情報を取得できます。

- **REQHDRS**　要求のロギング時に、ログファイルに要求ヘッダーも書き込みます。
- **RESHDRS**　要求のロギング時に、ログファイルに応答ヘッダーも書き込みます。
- **HEADERS**　要求のロギング時に、ログファイルに要求ヘッダーと応答ヘッダーの両方を書き込みます (REQHDRS、RESHDRS と同様)。

### デフォルト

ALL

### 備考

使用可能な値には、特定のメッセージタイプと HTTP ステータスコードを選択するキーワードがあります。カンマで区切って複数の値を指定できます。

### 参照


- 「LogFile (LOG) プロトコルオプション」388 ページ
- 「LogFormat (LF) プロトコルオプション」389 ページ
- 「LogMaxSize (LSIZE) プロトコルオプション」390 ページ


## MaxConnections (MAXCONN) プロトコルオプション

データベースサーバーで許可される同時接続の数を指定します。

### 構文

{ **MaxConnections** | **MAXCONN** }=*size*


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 使用可能なプロトコル

HTTP、HTTPS

### 指定可能な値

- **size**　この整数は、データベースサーバーで許可される同時接続の数を指定します。値 0 は、無制限であることを示します。

### デフォルト

5 (パーソナルサーバー)

ライセンスされている接続数 (ネットワークサーバー)

### 参照

- 「MaxRequestSize (MAXSIZE) プロトコルオプション」393 ページ
- 「サーバーライセンス取得ユーティリティ (dblic)」1003 ページ

## MaxRequestSize (MAXSIZE) プロトコルオプション

データベースサーバーで許可できる最大要求サイズを指定します。

### 構文

**{ MaxRequestSize | MAXSIZE }=***size***[ k | m | g ]**

### 使用可能なプロトコル

HTTP、HTTPS

### 指定可能な値

- **size**　この整数は、データベースサーバーで許可できる最大要求サイズを指定します。デフォルト値はバイト単位ですが、**k**、**m**、**g** のいずれかを使用してキロバイト、メガバイト、またはギガバイトの単位を指定できます。値 0 はこの制限を無効にしますが、細心の注意を払って使用してください。この制限がないと、悪質なクライアントがデータベースサーバーに過負荷をかけたり、メモリ不足を引き起こしたりする可能性があります。

### デフォルト

100k

### 備考

要求のサイズがこの制限を超えると、接続は閉じられ、クライアントは要求が大きすぎることを示す応答を返します。この値は要求サイズのみを制限するもので、応答サイズは制限しません。

### 参照

- 「MaxConnections (MAXCONN) プロトコルオプション」392 ページ

### 例

次のコマンドは、150000 バイトまでのサイズの要求を受け入れるようサーバーに指示します。

```
dbsrv12 -xs http{MaxRequestSize=150000}
```

## MyIP (ME) プロトコルオプション

クライアントで使用する場合は、クライアントがデータベースサーバーを検索するために使用するネットワークインターフェイスを指定します。データベースサーバーで使用する場合は、データベースサーバーが接続を受信するネットワークインターフェイスを指定します。

### 構文

{ **MyIP** | **ME** }={ *ip-address* [, *ip-address* ... ] | **NONE** }

### 使用可能なプロトコル

TCP/IP、HTTP、HTTPS

### 指定可能な値

- **ip-address**　この文字列は IP アドレス形式で指定する必要があります。複数の IP アドレスを指定する場合はカンマで区切ります。

- **NONE**　IP アドレスとしてキーワード NONE が指定されている場合は、アドレス情報を決定しようとしません。NONE キーワードは、複数のネットワークカードを持つコンピューターや、リモートアクセス (RAS) ソフトウェアとネットワークカードを持つコンピューターなど、この操作により大きな負荷がかかるコンピューター上のクライアントで使用するものです。データベースサーバーでは使用しないでください。

### 備考

MyIP (ME) プロトコルオプションは、複数のネットワークアダプターを持つコンピューターに対して指定します。

各アダプターには IP アドレスがあります。デフォルトでは、データベースサーバーは検出したすべてのネットワークインターフェイスを使用します。データベースサーバーがすべてのネットワークインターフェイスで受信しないようにする場合、使用する各インターフェイスのアドレスを MyIP (ME) プロトコルオプションで指定します。

Windows プラットフォームで IPv6 アドレスを指定する場合、インターフェイス識別子を使用する必要があります。UNIX プラットフォームでは、IPv6 アドレスのインターフェイス識別子とインターフェイス名の両方がサポートされます。Linux (カーネル 2.6.13 以降) では、インターフェイス識別子が必要です。

### 参照


- 「SQL Anywhere での IPv6 サポート」100 ページ
- 「TCP/IP プロトコル」99 ページ

### 例

次のコマンドライン (すべて 1 行に入力) は、サーバーに 2 つのネットワークカードを使用することを指示します。

```
dbsrv12 -x tcpip(MyIP=192.75.209.12,192.75.209.32) "%SQLANYSAMP12%¥demo.db"
```

次のコマンドライン (すべて 1 行に入力) は、IPv6 ネットワークカードを使用することをデータベースサーバーに指示します。

```
dbsrv12 -x tcpip(MyIP=fe80::5445:5245:444f) "%SQLANY12%¥demo.db"
```

次の接続文字列フラグメントは、クライアントがアドレス情報を決定しないように指定します。

```
LINKS=tcpip(MyIP=NONE)
```

## ReceiveBufferSize (RCVBUFSZ) プロトコルオプション

TCP/IP プロトコルスタックが使用するバッファーのサイズを設定します。

### 構文

{ **ReceiveBufferSize** | **RCVBUFSZ** }=*size*[ **k** | **m** | **g** ]

### 使用可能なプロトコル

TCP/IP

### 指定可能な値

- **size**　TCP/IP スタックが使用する受信バッファーのサイズを指定します。指定可能な最大サイズは 1 MB です。これは、その影響を熟知しているユーザーだけが使用するような高度なオプションです。

### デフォルト

コンピューターによって異なります。

### 備考

ネットワークに対する BLOB のパフォーマンスが重要な場合は、この値を増加できます。

### 参照

- 「TCP/IP プロトコル」99 ページ

## SendBufferSize (SNDBUFSZ) プロトコルオプション

TCP/IP プロトコルスタックが使用するバッファーのサイズを設定します。

### 構文

{ **SendBufferSize** | **SNDBUFSZ** }=*size*[ **k** | **m** | **g** ]

### 使用可能なプロトコル

TCP/IP

### 指定可能な値

- **size**　TCP/IP スタックが使用する送信バッファーのサイズを指定します。指定可能な最大サイズは 1 MB です。これは、その影響を熟知しているユーザーだけが使用するような高度なオプションです。

### デフォルト

コンピューターによって異なります。

### 備考

ネットワークに対する BLOB のパフォーマンスが重要な場合は、この値を増加できます。

### 参照

- 「TCP/IP プロトコル」99 ページ

## ServerPort (PORT) プロトコルオプション

データベースサーバーが実行されているポートを指定します。

### 構文

{ **ServerPort** | **PORT** }=*port-number*

### 使用可能なプロトコル

TCP/IP、HTTP、HTTPS

### 指定可能な値

- **port-number**　データベースサーバーの場合、ServerPort プロトコルオプションは TCP/IP を使用する通信のポート番号を指定します。

  単一のポート番号、または個々のポート番号の組み合わせやポート番号の範囲を指定できます。ポート番号のリストや範囲を指定すると、データベースサーバーは指定されたすべてのポート番号をバインドしようとします。次に例を示します。

  - (port=1234)
  - (port=1234,1235,1239)
  - (port=1234-1238)
  - (port=1234-1237,1239,1242)

  クライアントの場合、ServerPort (PORT) プロトコルオプションは、データベースサーバーが TCP/IP 通信を受信する 1 つまたは複数のポートをそのクライアントに知らせます。クライアントは、ServerPort (PORT) プロトコルオプションで指定されたすべてのポートにブロードキャストを実行して、データベースサーバーを検索します。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### デフォルト

- **TCP/IP** 2638
- **HTTP** 80
- **HTTPS** 443

### TCP/IP に関する注記

あるコンピューター上で TCP/IP を使用するすべてのアプリケーションは、それぞれ別の TCP/IP「ポート」を使用するので、ネットワークパケットは正しいアプリケーションに到達します。Internet Assigned Numbers Authority は、SQL Anywhere データベースサーバーに対して、TCP/IP 通信に使用するためのポート番号 2638 を割り当てています。ただし、その他のアプリケーションはこの予約ポートの使用を許可されていないわけではないため、データベースサーバーと別のアプリケーション間でアドレスが衝突する可能性があります。

- **サーバー側** サーバー側では、ServerPort (PORT) プロトコルオプションを指定した場合は、指定したポート番号のみが TCP/IP 接続に使用されます。UDP リスナーは、指定されたポート番号とデフォルトポート番号 (2638) の両方を使用します。ServerPort プロトコルオプションを指定しない場合は、デフォルトポート (2638) が使用されます。

- **クライアント側** ServerPort (PORT) プロトコルオプションを指定した場合は、指定したポート番号のみが TCP/IP 接続と UDP ブロードキャストに使用されます。ServerPort プロトコルオプションを指定しない場合は、ポート 2638 が使用されます。

  サーバーが 2638 以外のポートで実行中である場合、アプリケーションは、通常はサーバー名を指定し、ポート番号は指定しないでサーバーに接続できます。ポート番号は、データベースサーバーが Mac OS X 上で実行中であるときか、-sb サーバーオプションが使用されたときに指定する必要があります。

**ヒント**
Mac OS X では、同一の UDP ポートに複数のプロセスをバインドすることはできません。データベースサーバーがこれらのいずれかのプラットフォームで実行されている場合、指定の UDP ポート、またはポートの指定がないときはポート 2638 のみで受信します。

このため、データベースサーバーがデフォルトポート (2638) を使用しない場合は、クライアントで TCP/IP ポート番号を指定する必要があります。

たとえば、データベースサーバーがコマンド dbsrv12 -n MyServer "samples-dir/demo.db" によって起動される場合、同じサブネット上のクライアントは接続パラメーター Server=MyServer;LINKS=tcpip を使用してサーバーを見つけることができます。また別のサーバーが、コマンド dbsrv12 -n SecondServer "-x tcpip(PORT=7777) samples-dir/demo.db" によって Mac OS X で起動される場合は、同じサブネット上のクライアントは接続パラメーター Server=SecondServer;LINKS=tcpip(PORT=7777) を使用してサーバーを見つけることができます。データベースサーバーが Mac OS X 以外のプラットフォームで実行された場合は、クライアントが PORT パラメーターを指定する必要はありません。

さらに、Mac OS X では、SQL Anywhere データベースサーバーがポート 2638 をすでに使用していて、PORT プロトコルオプションなしで 2 番目のネットワークデータベースサーバーが起動された場合は、そのネットワークサーバーの起動は失敗します。この理由は、ユーザーはサーバーのポート番号を知っている必要があり、それを接続パラメーターで指定する必要があるからです。パーソナルサーバーの接続には、通常共有メモリが使用されるため、パーソナルサーバーは、ポート 2638 が使用中であっても正常に起動します。

### HTTP に関する注記

Web サーバーを使用している場合、デフォルトでは、データベースサーバーは標準 HTTP ポートと HTTPS ポートをそれぞれ 80 と 443 で受信します。

### 参照


- 「-x dbeng12/dbsrv12 サーバーオプション」 289 ページ
- 「-xs dbeng12/dbsrv12 サーバーオプション」 294 ページ
- 「-sb dbeng12/dbsrv12 サーバーオプション」 270 ページ


### 例

次の例は、PORT プロトコルオプションを使用する方法を示しています。

1. ネットワークデータベースサーバーを起動します。

   ```
   dbsrv12 -x tcpip -n server1
   ```

   ポート番号 2638 が取得されました。

2. 別のデータベースサーバーを起動しようとします。

   ```
   dbsrv12 -x tcpip -n server2
   ```

   デフォルトのポートは現在使用されているので、サーバーは別のポートを起動します ポート番号は、サーバーメッセージウィンドウに表示されます。Mac OS X の場合は失敗します。

3. 次のようにポート 2629 でデータベースサーバーを起動します。

```
dbsrv12 -x tcpip(ServerPort=2629) -n server3 c:¥mydata.db
```

4. 次のように、dbisql から、server3 が Host serverhost3 で起動され、Host 接続パラメーターを使って server3 に接続すると仮定します。

```
dbisql -c "UID=DBA;PWD=sql;ServerName=server3;Host=serverhost3:2629"
```

5. 次のように、dbisql から、server3 が Host serverhost3 で起動され、CommLinks 接続パラメーターを使って server3 に接続すると仮定します。

```
dbisql -c
"UID=DBA;PWD=sql;ServerName=server3;CommLinks=tcpip(Host=serverhost3;PORT=2629)"
```

この例では、コンピューター上の別の Web サーバーがすでにポート 80 を使用しているか、またはこのポート番号でサーバーを起動するためのパーミッションがユーザーにありません。8080 などの代替ポートで受信するサーバーを起動します。

```
dbsrv12 -xs http(PORT=8080) -n server3 web.db
```

## TDS プロトコルオプション (サーバー側のみ)

データベースサーバーへの TDS 接続が許可されるかどうかを制御します。

### 構文

**TDS={ YES | NO }**

### 使用可能なプロトコル

TCP/IP (サーバー側のみ)

### デフォルト

YES

### 戻り値

データベースサーバーへの TDS 接続を許可しないためには、TDS を NO に設定します。データベースサーバーに暗号化された接続だけを許可したい場合、TDS 接続を許可しないようにする方法はこのプロトコルオプションしかありません。

### 参照

- 「-ec dbeng12/dbsrv12 サーバーオプション」223 ページ

### 例

次のコマンドは、Sybase Open Client または jConnect アプリケーションからの接続は許可しないで、TCP/IP プロトコルを使ってデータベースサーバーを起動します。

```
dbsrv12 -x tcpip( TDS=NO ) ...
```

# Timeout (TO) プロトコルオプション

●通信確立時に、応答を待つ時間を秒単位で設定します (TCP/IP の場合)。

●要求を待つ時間を秒単位で設定します (HTTP または HTTPS)。

### 構文

{ **Timeout** | **TO** }=*timeout-value*

### 使用可能なプロトコル

TCP/IP、HTTP、HTTPS

### 指定可能な値

● **timeout-value**

○TCP/IP を使用するとき、この整数は、通信確立時に、応答を待つ時間を秒単位で設定します。指定可能な最長時間は、3600 秒です。

○HTTP または HTTPS を使用するときに、この整数は、完全な要求を受信するまで待つ時間の長さを秒単位で指定します。

### デフォルト

● **TCP/IP** 5

● **HTTP** 30

● **HTTPS** 30

### 戻り値

クライアントでは、このプロトコルオプションはどの程度長く待機するのかを指定します。

●サーバー検索時の UDP ブロードキャスト応答。

●データベースサーバーへの接続を行うための TCP/IP 接続呼び出し。

●切断時のデータベースサーバーからの応答。

サーバーで TCP/IP を使用しているときに、このプロトコルオプションは、どの程度の時間待機するかを指定します。

●ミラーリングまたは診断サーバーが接続を行うために検索するときの UDP ブロードキャスト応答。

●他のデータベースサーバーへのミラーリングまたは診断接続を行うときの TCP/IP 接続呼び出し。

●ミラーリングまたは診断サーバーから切断するときの応答。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

データベースサーバーに対して HTTP または HTTPS を使用しているときに、このプロトコルは次を指定します。

- 要求を受け取るときに許可されている最大アイドル時間。Timeout 期間は、要求が開始するときに開始し、要求が完了するときに終了します。この制限に達した場合は、接続が閉じられて要求タイムアウトメッセージがクライアントに返されます。

### 参照

- 「KeepaliveTimeout (KTO) プロトコルオプション」386 ページ

### 例

次の接続文字列は、クライアントライブラリに対して、20 秒のタイムアウトで TCP/IP を使用してデータベースサーバーに接続するように指示します。

```
...
CommLinks=tcpip(TO=20)
...
```

## trusted_certificate プロトコルオプション

信頼できる証明書を 1 つ以上含むファイルのパスとファイル名を指定します。

### 構文

**trusted_certificate=***public-certificate*

### 使用可能なプロトコル

TLS、HTTPS

### デフォルト

なし

### 備考

クライアントは、trusted_certificate 暗号化プロトコルオプションを使用して、信用されたデータベースサーバー証明書を指定します。信頼できる証明書は、サーバーの自己署名証明書、パブリックエンタープライズルート証明書、民間認証局に属する証明書のいずれかです。FIPS 認定の RSA 暗号化を使用している場合は、RSA を使用して証明書を生成する必要があります。Encryption 接続パラメーターで TLS を指定する場合、このプロトコルオプションは必須です。

HTTPS は、Web サービスのクライアントプロシージャーだけでサポートされています。「CREATE PROCEDURE 文 [Web サービス]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

**注意**
別途ライセンスが必要な必須コンポーネント

ECC 暗号化と FIPS 認定の暗号化には、別途ライセンスが必要です。強力な暗号化テクノロジはすべて、輸出規制対象品目です。

「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

### 参照


- 「certificate_company プロトコルオプション」374 ページ
- 「certificate_name プロトコルオプション」375 ページ
- 「certificate_unit プロトコルオプション」376 ページ
- 「Encryption (ENC) 接続パラメーター」342 ページ
- 「SQL Anywhere クライアント／サーバー通信の暗号化」1420 ページ
- 「証明書作成ユーティリティ (createcert)」924 ページ
- 「-xs dbeng12/dbsrv12 サーバーオプション」294 ページ


### 例

次のコマンドは、トランスポートレイヤーセキュリティを使用して SQL Anywhere のサンプルデータベースを Interactive SQL に接続します。

```
dbisql -c
"UID=DBA;PWD=sql;HOST=myhost;Server=demo;ENC=TLS(
tls_type=RSA;FIPS=n;trusted_certificate=c:¥temp¥myident;
certificate_unit='SA';certificate_company='Sybase iAnywhere';
certificate_name='Sybase')"
```

## VerifyServerName (VERIFY) プロトコルオプション (クライアント側のみ)

クライアントが接続前にデータベースサーバー名を確認する必要があるかどうかを制御します。

### 構文

**{ VerifyServerName | VERIFY }={ YES | NO }**

### 使用可能なプロトコル

TCP/IP (クライアント側のみ)

### デフォルト

YES

### 戻り値

TCP を介して接続している場合、DoBroadcast=NONE パラメーターを指定すると、クライアントによって TCP 接続が行われ、検出されたサーバーと検索対象サーバーの名前が一致しているかどうかが検証されます。VerifyServerName=NO を指定すると、サーバー名は検証されません。そ

のため、IP アドレスかポートさえわかっていれば、SQL Anywhere クライアントから SQL Anywhere サーバーに接続できます。

この場合も接続文字列にサーバー名を指定する必要がありますが、その指定は無視されます。VerifyServerName (VERIFY) プロトコルオプションは、DoBroadcast=NONE を指定した場合にのみ使用します。

サーバーが -sb 0 または BroadcastListener=NO を使用している場合、クライアントはサーバーに接続するために DoBroadcast=NONE を指定する必要はありません。ただし、HOST= を指定することは必要です。dblocate ユーティリティはサーバーを検索しません。

**注意**
このパラメーターは、各サーバーにユニークなサーバー名を付けることができないなど、特別な場合にのみ使用してください。接続するときは、ユニークなサーバー名を使用することをおすすめします。サーバーへの接続に最適な方法は、各サーバーにユニークなサーバー名を付け、その名前を使用することです。

**参照**

● 「DoBroadcast (DOBROAD) プロトコルオプション」379 ページ

# SQL Anywhere for Windows Mobile

SQL Anywhere では、クライアントアプリケーションとサーバーアプリケーションの両方を Windows Mobile デバイスに配備できます。この配備モデルによって、データベースアプリケーションは自律的に実行できます。SQL Anywhere は、Windows Mobile デバイスで実行されるネットワークデータベースサーバーを提供します。ネットワークデータベースサーバーは、TCP/IP 経由の通信をサポートします。Windows Mobile はネットワークデータベースサーバーをサポートするため、デスクトップコンピューターで管理ユーティリティを使用して、Windows Mobile データベースに対するタスクを実行できます。次に例を示します。

●デスクトップコンピューターで Sybase Central を使用してデータベースを管理できる

●デスクトップコンピューター上の Interactive SQL を使用して、データのロードとアンロード、クエリを実行できる

**参照**

● 「Windows Mobile の管理ユーティリティ」427 ページ
● 「チュートリアル：Sybase Central から Windows Mobile データベースを実行する」427 ページ

# Windows Mobile デバイスへの SQL Anywhere のインストール

### 稼働条件

- サポートされている Windows オペレーティングシステムを実行しているコンピューター
  - Windows Vista 以降の場合：Windows Mobile Device Center 6.1 以降
  - Windows XP 以前の場合：Microsoft ActiveSync 3.5 以降
- SQL Anywhere がサポートしている Windows Mobile デバイス

  SQL Anywhere でサポートされている Windows Mobile デバイスのリストについては、http://www.ianywhere.jp/sas/os.html を参照してください。

### Windows Mobile ファイルのロケーション

Windows Mobile における SQL Anywhere インストールディレクトリは、デバイスのタイプとインストール先のロケーションによって異なります。サブフォルダーは作成されません。すべての DLL は、*¥Windows* ディレクトリにインストールされます。

| ロケーション | インストールディレクトリ |
|---|---|
| ストレージカード | *¥storage-card¥Sybase SQL Anywhere 12* |
| ストレージカード | *¥storage-card¥SQLAny12* |
| メインメモリ | *¥Program Files¥SQLAny12* |

**警告**
SQL Anywhere をストレージカード (SD カードなど) にインストールしないことをおすすめします。

Windows Mobile デバイスが中断状態から再開するとき、リムーバブルデバイスにあるすべての開いているファイル (実行プログラムと DLL を含む) がオペレーティングシステムによって閉じられる場合があります。オペレーティングシステムでは、デバイスが中断されたときに実行されていたプログラムによって使用中であった実行プログラムと DLL へのアクセスが失われます。この場合、オペレーティングシステムは、エラーメッセージを表示しないでプロセス (SQL Anywhere データベースサーバーなど) を終了する可能性があります。

# インストール時の考慮事項：Windows Mobile での ICU の使用

Unicode 照合アルゴリズム (UCA) は、Unicode 文字セット全体のソートに使用するアルゴリズムです。これにより、言語的に正しい比較、ソート、大文字小文字変換が実現されます。UCA は Unicode 規格の一部として開発されました。SQL Anywhere では、IBM が開発および保守してい

る International Components for Unicode (ICU) オープンソースライブラリを使用して UCA を実装しています。

Windows Mobile では、NCHAR 照合または CHAR 照合として UCA を使用している場合は、ICU が必要です。また、CHAR 文字セットがオペレーティングシステムの文字セットと一致しない場合も、Windows Mobile で ICU が必要です。

デフォルトでは、ICU ライブラリは Windows Mobile にインストールされません。これは、このライブラリをインストールすると、Windows Mobile に SQL Anywhere をインストールするときのサイズが約 1.7 MB 大きくなるからです。ただし、ICU ライブラリが必要な場合は SQL Anywhere インストールを変更できます。

ICU ライブラリをインストールしていない場合は、Windows Mobile の文字セットと一致する文字セットを使う照合を選択するか、データベースを作成するときに CHAR 照合として UTF8BIN 照合を選択することが必要です。データベースを作成するときに NCHAR 照合として UTF8BIN を選択することも必要になります。

**注意**
デスクトップにデータベースを作成して Windows Mobile デバイスに配備するときは、Windows Mobile デバイスに ICU ライブラリがインストールされている場合のみ UCA 照合を使用できます。ICU ライブラリがインストールされていない場合、UCA を使うデータベースを Windows Mobile で利用することはできません。

**参照**


- 「Unicode 照合アルゴリズム (UCA)」489 ページ
- 「文字セット変換」478 ページ


## インストール時の考慮事項：Windows Mobile での .NET Compact Framework の使用

API の最新バージョンは ADO.NET 3.5 ですが、SQL Anywhere がサポートする大部分のデバイスには ADO.NET 1.x サポートのみがインストールされています。デバイスで ADO.NET バージョン 2.0 または 3.5 を使用するには、Microsoft 社の Web サイトから ADO.NET 2.0 または 3.5 のサポートをダウンロードし、デバイスにインストールする必要があります。

- **バージョン 2.0**　ADO.NET 2.0 を使用したアプリケーション開発の詳細については、「SQL Anywhere .NET データプロバイダー」『SQL Anywhere サーバー プログラミング』と「ネームスペース」『SQL Anywhere サーバー プログラミング』を参照してください。

- **バージョン 3.5**　ADO.NET 3.5 を使用したアプリケーション開発の詳細については、「SQL Anywhere .NET データプロバイダー」『SQL Anywhere サーバー プログラミング』と「ネームスペース」『SQL Anywhere サーバー プログラミング』を参照してください。

**参照**


- 「SQL Anywhere .NET データプロバイダーの使用」『SQL Anywhere サーバー プログラミング』

## インストール時の考慮事項 : Windows Mobile 5.0 for Smartphone に関する制限

SQL Anywhere サーバーの Windows Mobile 用の機能はすべてスマートフォンでサポートされていますが、Windows Mobile 5.0 には次の制限があります。

- **共有メモリプロトコルがサポートされない**　通信プロトコルを指定しなくても、TCP/IP が使用されます。接続を確立するときはデータベースサーバー名を指定する必要があります。指定しないと、接続が失敗します。

- **[サーバー起動オプション] ウィンドウがサポートされない**　データベースサーバーを起動し、オプションを何も指定しないと、**[サーバー起動オプション]** ウィンドウが表示されます。不完全なコマンドや不適切なコマンドを指定してデータベースサーバーを起動しようとすると、エラーが表示され、データベースサーバーは起動されません。

- **ODBC 接続で接続情報が要求されないことがある**　SQLDriverConnect で ODBC DriverCompletion パラメーターを使用すると、追加の接続情報を要求するプロンプトが表示されることがあります。このプロンプトが表示されません。SQLDriverConnect に失敗した場合、プロンプトが表示されず、エラーが返されます。

- **アンロード／再ロードがサポートされない**　別のプラットフォームで Windows Mobile のデータベースを再構築し、Windows Mobile デバイスにデータベースをコピーする必要があります。これは、Windows Mobile データベースを再構築するときに推奨される方法です。

  次の項を参照してください。

  ○「Windows Mobile でのデータベースの再構築」423 ページ
  ○「データベースの再構築」『SQL Anywhere サーバー SQL の使用法』

## SQL Anywhere for Windows Mobile のインストール

**♦ Windows Mobile デバイスに SQL Anywhere for Windows Mobile をインストールするには、次の手順に従います。**

1. サポートされている Windows オペレーティングシステムを実行しているデスクトップコンピューターに、Windows Mobile デバイスを接続します。

2. **[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[Windows Mobile に展開]** をクリックします。

3. **SQL Anywhere for Windows Mobile Deployment ウィザード**の指示に従います。

# Windows Mobile サンプルアプリケーション

サンプルアプリケーションで使用されるサンプルデータベースは、*demo.db* という名前のファイルで、Windows Mobile デバイスの *¥My Documents* フォルダーにあります。Windows バージョン

の SQL Anywhere に付属のサンプルデータベースとは異なり、このサンプルデータベースには International Components for Unicode (ICU) ライブラリは必要ありません。その他のすべての点については、データベースは Windows バージョンと同じです。

SQL Anywhere for Windows Mobile のインストールには、次のサンプルアプリケーションが含まれています。

- ADO.NET Sample
- ESQL Sample
- ODBC Sample

これらのアプリケーションを使用すると、サンプルデータベースにアクセスして、SQL Anywhere for Windows Mobile の機能を検証できます。

**参照**

- 「インストール時の考慮事項：Windows Mobile での ICU の使用」404 ページ

## Windows Mobile サンプルアプリケーションの起動

SQL Anywhere サーバーのアイコンを使用してデータベースサーバーを起動します。

**♦ SQL Anywhere データベースサーバーを起動するには、次の手順に従います。**

1. **[スタート]** » **[プログラム]** » **[SQLAny12]** の順にタップします。
2. **[Server]** をタップします。
3. **[データベース]** フィールドに *¥My Documents¥demo.db* と入力するか、**[参照]** をタップして *My Documents* ディレクトリの *demo.db* を探します。
4. **[サーバー名]** フィールドは空白のままにしてもかまいません。

   サーバー名はデフォルトでデータベース名 *demo* になります。
5. **[キャッシュ]** フィールドは変更しないでそのままにしてもかまいません。
6. **[オプション]** フィールドは空白のままにしてもかまいません。
7. **[OK]** をタップして、データベースサーバーで実行中のサンプルデータベースを開始します。
8. デバイスの **[Today]** 画面に移動します。
9. 画面の右下にあるデータベースサーバーアイコンをタップします。

   データベースサーバーメッセージウィンドウに「**要求を受信中です。**」というメッセージが表示されたら、次のレッスンに進むことができます。

サンプルデータベースを使い終わったら、データベースサーバーを停止してください。

♦ **データベースサーバーを停止するには、次の手順に従います。**

1. [Today] 画面の右下にあるネットワークデータベースサーバーアイコンをタップします。
2. **[メニュー]** » **[シャットダウン]** をタップします。

## ADO.NET Sample の使用

♦ **ADO.NET Sample の使用**

ADO.NET Sample は、ADO.NET プログラミングインターフェイスを使用する簡単なアプリケーションのデモです。このアプリケーションでは、ネットワークデータベースサーバーで実行中のサンプルデータベースを起動し、SQL 文でアクセスして、データを変更できます。

ADO.NET Sample を使用するには、デバイスに Microsoft .NET Compact Framework バージョン 2.0 または 3.0 をインストールしておく必要があります。Windows Mobile 6 デバイスには Microsoft .NET Compact Framework バージョン 2.0 がインストールされていますが、Windows Mobile 5 デバイスにはインストールされていません。ADO.NET Sample は、タッチスクリーンを備えた Windows Mobile Classic および Professional のデバイスのみをサポートしています。

このサンプルのソースコードは、*%SQLANYSAMP12%¥SQLAnywhere¥ce¥ado_net_sample* にあります。

Microsoft Visual Studio では、*%SQLANYSAMP12%¥SQLAnywhere¥ce¥ado_net_sample ¥ado_net_sample.sln* からこのプロジェクトをロードできます。

このコンポーネントを Microsoft ダウンロードセンターからダウンロードするには、次の URL にアクセスしてください。http://www.microsoft.com/downloads/search.aspx?displaylang=en

> **注意**
> ADO.NET Sample のユーザーインターフェイスでは、SQL 文を 1 行で入力する必要があります。

1. **[スタート]** » **[プログラム]** » **[SQLAny12]** » **[ADO.NET Sample]** の順にタップします。
2. **[Connect]** をタップします。

   SQL Anywhere データベースサーバーを先に起動しなかった場合は、**[接続]** をタップするとサーバーが起動します。
3. **[Exec SQL]** をタップして、デフォルトの SQL 文 **SELECT * FROM Employees** を実行します。

   Employees テーブルのデータが、データウィンドウに表示されます。
4. データウィンドウの右と下にあるスクロールバーを使って、Employees テーブルのデータ間を移動します。
5. 次のような、特定のデータ範囲にアクセスするクエリを入力します。

   ```
   SELECT EmployeeID, Surname FROM Employees;
   ```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

6. **[Exec SQL]** をタップして、SQL 文を実行します。

   指定したデータ範囲が、データ枠の元のデータを置き換えます。

7. **SELECT * FROM Employees ORDER BY EmployeeID** と入力し、**[Exec SQL]** をタップします。

   EmployeeID が 105 の従業員 Matthew Cobb に注目します。

8. **UPDATE Employees SET Surname = 'Jones' WHERE Surname = 'Cobb'** と入力し、**[Exec SQL]** をタップして SQL 文を実行します。

9. **SELECT * FROM Employees ORDER BY EmployeeID** と入力し、**[Exec SQL]** をタップします。

   Matthew の姓が Cobb から Jones に変わりました。

10. **UPDATE Employees SET Surname = 'Cobb' WHERE Surname = 'Jones'** と入力し、**[Exec SQL]** をタップして、サンプルデータベースに加えた変更を元に戻します。

11. **SELECT * FROM Employees ORDER BY EmployeeID** と入力し、**[Exec SQL]** をタップして、内容が元に戻ったことを確認します。

    Matthew の姓が Jones から元の Cobb に戻りました。

12. **SELECT * FROM Customers** と入力し、**[Exec SQL]** をタップして、別のテーブルのデータにアクセスします。

    Customers テーブルのすべてのデータがデータウィンドウに表示されて、Employees テーブルのデータを置き換えます。

13. **[Disconnect]** をタップして、データベースサーバーを切断します。

    ADO.NET Sample が切断されます。接続はプールされるため、データベースサーバーは自動的には停止しません。

14. ウィンドウの右上隅の **[X]** をタップして、ADO.NET Sample を閉じます。

## ESQL Sample の使用

**♦ ESQL Sample を使用するには、次の手順に従います。**

ESQL Sample は、Embedded SQL プログラミングインターフェイスを使用する簡単なアプリケーションのデモです。このアプリケーションでは、ネットワークデータベースサーバーで実行中のサンプルデータベースを起動し、SQL 文を使用してデータにアクセスできます。

このサンプルのソースコードは、*%SQLANYSAMP12%¥SQLAnywhere¥ce¥esql_sample* にあります。

Visual Studio 2005 では、*%SQLANYSAMP12%¥SQLAnywhere¥ce¥esql_sample¥esql_sample.sln* からこのプロジェクトをロードできます。

**注意**
ESQL Sample のユーザーインターフェイスでは、SQL 文を 1 行で入力する必要があります。

1. **[スタート]** » **[プログラム]** » **[SQLAny12]** » **[ESQL Sample]** の順にタップして、ESQL Sample を開始します。
2. **[Connect]** をタップし、デフォルトの接続文字列を使ってサンプルデータベースに接続します。
3. **[Exec SQL]** をタップして、デフォルトの SQL 文 **SELECT * FROM Employees** を実行します。

   Employees テーブルのデータが、データウィンドウに表示されます。
4. Employee テーブルのデータを見るには、スクロールバーを使用します。
5. Customers テーブルのデータにアクセスするには、**SELECT * FROM Customers** と入力して、**[Exec SQL]** をタップします。

   データウィンドウで、Customer のデータが Employee のデータを置き換えます。
6. **[Menu]** » **[Disconnect]** をタップして、ネットワークデータベースサーバーを停止します。

   ESQL Sample が切断され、ネットワークデータベースサーバーが停止します。
7. ウィンドウの右上隅の **[X]** をタップして、ESQL Sample を閉じます。

## ODBC Sample の使用

**♦ ODBC Sample を使用するには、次の手順に従います。**

ODBC Sample は、ODBC プログラミングインターフェイスを使用する簡単なアプリケーションのデモです。このアプリケーションでは、ネットワークデータベースサーバーで実行中のサンプルデータベースを起動し、基本的な SQL 文を使用してデータにアクセスできます。

このサンプルのソースコードは、*%SQLANYSAMP12%¥SQLAnywhere¥ce¥odbc_sample* にあります。

Visual Studio 2005 では、*%SQLANYSAMP12%¥SQLAnywhere¥ce¥odbc_sample¥odbc_sample.sln* からこのプロジェクトをロードできます。

**注意**
ODBC Sample のユーザーインターフェイスでは、SQL 文を 1 行で入力する必要があります。

1. **[スタート]** » **[プログラム]** » **[SQLAny12]** » **[ODBC Sample]** の順にタップして、ODBC Sample を開始します。
2. **[Connect]** をタップします。
3. **[Exec SQL]** をタップして、デフォルトの SQL 文 **SELECT * FROM Employees** を実行します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Employees テーブルのデータが、データウィンドウに表示されます。

4. Employee テーブルのデータを見るには、スクロールバーを使用します。

5. Customers テーブルのデータにアクセスするには、**SELECT * FROM Customers** と入力して、**[Exec SQL]** をタップします。

   データウィンドウで、Customer のデータが Employee のデータを置き換えます。

6. **[Menu]** » **[Disconnect]** をタップして、ネットワークデータベースサーバーを停止します。

   ODBC Sample が切断され、ネットワークデータベースサーバーが停止します。

7. ODBC Sample を閉じます。

## Windows Mobile デバイスで実行中のデータベースへの接続

デスクトップコンピューター上で実行しているアプリケーションを、Windows Mobile デバイス上で実行しているデータベースに接続するには、デスクトップコンピューターと Windows Mobile デバイス間の Windows Mobile Device Center (ActiveSync) リンクを介した TCP/IP 接続によって実現できます。これにより、デスクトップコンピューター上の管理ユーティリティを使用して、Windows Mobile データベースを管理できます。

**参照**

- 「Windows Mobile の管理ユーティリティ」427 ページ

## Windows Mobile デバイス上でのデータベースサーバーの起動

**♦ リモート接続用に、Windows Mobile でデータベースサーバーを起動するには、次の手順に従います。**

Windows Mobile で実行されているデータベースサーバーにデスクトップコンピューターから接続するには、サーバーを起動するときに TCP/IP オプションを選択する必要があります。

1. **[スタート]** » **[プログラム]** » **[SQLAny12]** の順にタップします。

2. **[Server]** をタップします。

3. 起動するデータベースファイルの名前を **[データベース]** フィールドに入力するか、**[参照]** をクリックしてデータベースを指定します。

   サンプルデータベースはデフォルトで *¥My Documents¥demo.db* にあります。

4. **[サーバー名]** フィールドに、使用するデータベースサーバー名を入力します。

   サンプルデータベースサーバーのデフォルト名は 「demo」 です。

5. **[オプション]** フィールドに **-x tcpip(port=2639)** と入力します。

   データベースサーバーが使用するポートは 2639 です。ポート 2638 を使用することはおすすめしません。このポートは、デスクトップシステムで実行されている SQL Anywhere サーバーのデフォルト TCP/IP ポートでもあるためです。詳細については、「-x dbeng12/dbsrv12 サーバーオプション」289 ページを参照してください。

6. **[TCP/IP を使用]** をクリックします。

   Windows Mobile デバイスで実行中のデータベースにデスクトップコンピューターから接続するには、TCP/IP 接続が必要です。

7. **[OK]** をタップして、ネットワークデータベースサーバーで実行中のサンプルデータベースを開始します。

これで、デスクトップコンピューターから Windows Mobile デバイスに接続するための ODBC データソースを作成できます。

**参照**

- 「Windows Mobile デバイスに接続するための ODBC データソースの作成」412 ページ

## Windows Mobile デバイスに接続するための ODBC データソースの作成

**♦ Windows Mobile デバイスで稼働しているデータベースに接続するための ODBC データソースを作成するには、次の手順に従います。**

1. Windows デスクトップコンピューターで、**[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[ODBC データソースアドミニストレーター]** をクリックします。

2. **[ユーザー DSN]** タブで、**[追加]** をクリックします。

3. **[SQL Anywhere 12]** をクリックして、**[完了]** をクリックします。

4. **[ODBC]** タブの **[データソース名]** フィールドに、**MobileServer** と入力します。

5. **[ログイン]** タブをクリックします。
   - **[認証]**　**[データベース]** をクリックします。
   - **[ユーザー ID]**　このフィールドが空白であることを確認します。データベースに接続するたびに、ユーザー ID とパスワードを指定する必要があります。
   - **[パスワード]**　このフィールドが空白であることを確認します。**[パスワードの暗号化]** ボックスはオフにします。
   - **[アクション]**　**[別のコンピューターで稼働しているデータベースに接続]** をクリックします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

- **[ホスト]**　USB 接続を介して Windows Mobile Device Center (ActiveSync) を使用して Windows Mobile 5.0 または 6.0 デバイスに接続した場合は、Windows Mobile デバイスの IP アドレス **169.254.2.1** を入力してみてください。

  一部の Windows Mobile デバイス (特に古いデバイス) では、このアドレスを使用すると問題が発生することがあります。その場合は、localhost (**127.0.0.1**) IP アドレスを試してください。この場合、プロキシポートも設定します。プロキシポートの設定の詳細については、「Windows Mobile デバイス用プロキシポートの作成」413 ページを参照してください。

  詳細については、「Windows Mobile デバイスの IP アドレスの特定」415 ページを参照してください。

- **[ポート]**　Windows Mobile デバイスが受信するポート番号を指定します。

  Windows Mobile デバイス上のデータベースサーバーが起動時に使用するよう設定されているポートを使用します。ポート **2639** を使用することをおすすめします。「Windows Mobile デバイス上でのデータベースサーバーの起動」411 ページを参照してください。

- **[サーバー名]**　Windows Mobile デバイスで実行されている特定のデータベースサーバーに接続する場合は、サーバー名を入力します。それ以外の場合は、このフィールドが空白であることを確認します。

- **[データベース名]**　Windows Mobile デバイスで実行されている特定のデータベースに接続する場合は、データベース名を入力します。それ以外の場合は、このフィールドが空白であることを確認します。

6. **[OK]** をクリックします。

これで、このデータソースを使用して、Windows Mobile デバイス上で実行中のデータベースに Windows デスクトップコンピューターから接続できます。

接続をテストするには、dbping ユーティリティを使用できます。データベースが Windows Mobile デバイス上で起動されている場合は、次のようなコマンドを使用して接続をテストします。

```
dbping -c "DSN=MobileServer;UID=DBA;PWD=sql"
```

**参照**


- 「ODBC データソース」36 ページ
- 「Windows Mobile の管理ユーティリティ」427 ページ


# Windows Mobile デバイス用プロキシポートの作成

デスクトップから Windows Mobile デバイス上のデータベースサーバーに接続するには、localhost (**127.0.0.1**) IP アドレスへの要求が Windows Mobile デバイス上で実行中のデータベースに渡されるように、Mobile Device Center (ActiveSync) のプロキシポートを設定することが必要になる場合があります。

**♦ Windows Mobile デバイス用プロキシポートを作成するには、次の手順に従います (Interactive SQL の場合)。**

1. デスクトップコンピューターで Interactive SQL を起動します。32 ビットバージョンの Windows Mobile Device Center (ActiveSync) を使用する場合は、32 ビットバージョンの Interactive SQL を起動します。

2. **[接続]** ウィンドウの下部で、**[ツール]**、**[Windows Mobile プロキシポートの設定]** をクリックします。

3. **[新規]** をクリックします。

4. **[名前]** フィールドに **SQL Anywhere** と入力します。

5. **[ポート]** フィールドに、**2639** と入力します。

   Windows Mobile 上で実行中の SQL Anywhere データベースサーバーにはこの TCP/IP ポートを使用することをおすすめします。Windows Mobile デバイス上のデータベースサーバーが同じポートを使用するように設定されている場合にのみ、他のポートを使用できます。

   **注意**
   Windows Mobile デバイスをクレードルに置くたびに、Windows Mobile Device Center (ActiveSync) がポート 2639 のトラフィックをデバイスに転送します。Windows Mobile デバイスがクレードルにある間にデスクトップコンピューターでデータベースサーバーを起動すると、ポート 2639 を使用できません。これが問題になる場合は、Windows Mobile トラフィック専用に別のポートを選択することもできます。

6. **[OK]** をクリックします。

   プロキシポートの設定や変更を行うとき、プロキシポートの新しい設定を有効にするために Windows デスクトップをログオフして、もう一度ログオンすることが必要になる場合があります。

レジストリキー **HKEY_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥Windows CE Services ¥ProxyPorts** を設定して、プロキシポートを設定することもできます。DWORD 値は **SQL Anywhere** とし、Windows Mobile トラフィック専用のポートを指定します (2639 を選択することをおすすめします)。64 ビットバージョンの Windows で 32 ビットバージョンの Windows Mobile Device Center (ActiveSync) を実行している場合、このレジストリエントリは 32 ビットのハイブに格納されます。

**警告**
レジストリの変更は危険です。レジストリは、ユーザー自身の責任で変更してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

# Windows Mobile デバイスの IP アドレスの特定

### ♦ Windows Mobile デバイスの IP アドレスの特定

Windows Mobile で実行中のデータベースへの接続を確立するには、IP アドレスが必要になることがあります。

1. **[スタート]** » **[プログラム]** » **[SQLAny12]** の順にタップします。
2. **[サーバー]** をタップします。
3. 起動するデータベースファイルの名前を **[データベース]** フィールドに入力するか、**[参照]** をクリックしてデータベースを指定します。

   サンプルデータベースはデフォルトで *¥My Documents¥demo.db* にあります。
4. **[サーバー名]** フィールドに、使用するデータベースサーバー名を入力します。

   サンプルデータベースサーバーのデフォルト名は 「demo」 です。
5. **[オプション]** フィールドに **-z -x tcpip(port=2639)** と入力します。

   データベースサーバーは -z オプションを使用して、起動中に IP アドレスを表示します。Windows Mobile デバイスをネットワークから切断して再接続すると、アドレスが変わることがあります。詳細については、「-z dbeng12/dbsrv12 サーバーオプション」296 ページを参照してください。

   データベースサーバーが使用するポートは 2639 です。ポート 2638 を使用することはおすすめしません。このポートは、デスクトップシステムで実行されている SQL Anywhere サーバーのデフォルト TCP/IP ポートでもあるためです。詳細については、「-x dbeng12/dbsrv12 サーバーオプション」289 ページを参照してください。
6. **[TCP/IP を使用]** をクリックします。

   Windows Mobile デバイスで実行中のデータベースにデスクトップコンピューターから接続するには、TCP/IP 接続が必要です。
7. **[OK]** をタップして、ネットワークデータベースサーバーで実行中のサンプルデータベースを開始します。
8. デバイスの **[Today]** 画面に移動します。
9. 画面の右下にあるデータベースサーバーアイコンをタップします。

   IP アドレスはデータベースサーバーメッセージウィンドウに表示されます。リストの最初の「**IP アドレス %1!s! で UDP リスナーを開始しています。**」メッセージで IP アドレスを確認します。

**参照**

- 「Windows Mobile デバイスに接続するための ODBC データソースの作成」412 ページ

# Windows Mobile データベースサーバーの設定

SQL Anywhere の完全バージョンで使用できる SQL 機能のほとんどは、Windows Mobile バージョンでも使用できます。たとえば、トランザクション処理、参照整合性アクション、プロシージャー、トリガーなどです。ただし、Java 機能とリモートデータアクセス機能は、Windows Mobile では使用できません。

Windows Mobile デバイスで使用するデータベースを作成するときは、サポートされていない機能に注意してください。機能の完全なリストについては、「Windows Mobile での SQL Anywhere 機能のサポート」434 ページを参照してください。

以下の設定は、データベースの作成中に設定します。いったん設定すると、データベースを再構築しなければ変更できません。

- **大文字と小文字の区別の有無**　「大文字と小文字の区別」『SQL Anywhere サーバー SQL の使用法』を参照してください。
- **比較における後続ブランクの処理**　デフォルトでは、後続ブランクを余分の文字と見なしてデータベースが作成されます。たとえば、'Dirk' は 'Dirk ' と同じではありません。後続ブランクが無視されるように、ブランクを埋め込んだデータベースを作成できます。「比較の後続ブランクを無視」『SQL Anywhere サーバー SQL の使用法』を参照してください。
- **ページサイズ**　「適切なページサイズの使用」『SQL Anywhere サーバー SQL の使用法』を参照してください。
- **照合順と文字セット**　Windows Mobile 用のデータベースを作成するときは、該当言語のために Windows で使用されるシングルバイトまたはマルチバイトの文字セットと同じ文字セットを基にした照合を使用してください。たとえば、英語、フランス語、またはドイツ語を使用する場合は、1252Latin1 照合を使用します。日本語を使用する場合は 932JPN 照合を、韓国語を使用する場合は 949KOR 照合を使用します。「照合」487 ページを参照してください。

> **注意**
> Windows Mobile で使用するデータベースを作成するときは、適合化オプションの文字列にロケールまたはソートタイプを指定しないでください。指定すると、Windows Mobile デバイスでデータベースが起動しなくなる可能性があります。 照合の適合化オプションの詳細については、「照合の適合化オプション」496 ページを参照してください。

文字セット変換は Windows Mobile ではサポートされていないため、Windows Mobile 上のデータベースには、オペレーティングシステムの文字セットか UTF-8 のいずれかを使用する必要があります。

Windows Mobile データベースを作成するときに、ICU ライブラリをインストールするかどうかを選択する必要があります。「インストール時の考慮事項：Windows Mobile での ICU の使用」404 ページを参照してください。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## Windows Mobile のトランザクションログ

トランザクションログには、データベースに加えられた変更がその変更順に格納されます。データベースファイルでメディア障害が発生した場合、トランザクションログはデータベースのリカバリに重要な役割を果たします。また、トランザクションログを使用すると、より効率的に作業できます。トランザクションログは、デフォルトではデータベースファイルと同じディレクトリに配置されます。このログは、Windows Mobile デバイスで初めてデータベースを開始するときに作成されます。

既存のデータベースを Windows Mobile デバイスにコピーする場合は、データベースファイルとトランザクションログファイルの両方をコピーできます。トランザクションログファイルをデバイスにコピーしなければ、Windows Mobile デバイスでデータベースを開始したときに新しいログが作成されます。新しいトランザクションログには、元のトランザクションログに格納されていた情報は含まれません。これは、最後にデータベースを使用したときに正しく停止しなかった場合や、データベースを同期している場合に問題になることがあります。データベースファイルとトランザクションログファイルの両方を Windows Mobile デバイスにコピーすることをおすすめします。

**参照**

- 「トランザクションログ」127 ページ

## Windows Mobile の jConnect

Sybase jConnect は、SQL Anywhere で使用できる純正 Java JDBC ドライバーです。Sybase Central と Interactive SQL には、jConnect JDBC ドライバーを使用して SQL Anywhere のデータベースにアクセスできるように、jConnect メタデータサポートを追加するオプションがあります。

Windows Mobile 用に作成されているデータベースの場合、デフォルトでは jConnect メタデータサポートが**データベース作成ウィザード**によって有効にされません。ただし、必要に応じて、jConnect メタデータサポートを有効にすることができます。

データベースに jConnect メタデータサポートを追加すると、多数のエントリがシステムテーブルに追加されます。これによってデータベースサイズが大きくなり、たとえ jConnect 機能を使用しない場合でも、データベースの実行に必要なメモリが約 200 KB 増えます。

Windows Mobile のような制限されたメモリ環境で作業をする場合、jConnect を使用する予定がなければ、データベースに jConnect メタデータサポートを追加しないことをおすすめします。

**参照**

- 「jConnect JDBC ドライバー」『SQL Anywhere サーバー プログラミング』

## Windows Mobile の暗号化

データベースを安全に管理するために、単純暗号化または強力な暗号化のいずれかを選択できます。データベースを初期化した後で暗号化の設定を変更するには、データベース全体を再構築するしかありません。

**参照**


- 「データベースの暗号化と復号化」1396 ページ
- 「Windows Mobile データベースセキュリティ」1408 ページ


## Windows Mobile データベースサーバーの作成

Windows Mobile デバイス用の SQL Anywhere データベースは、次の方法で作成できます。

- Sybase Central の**データベース作成ウィザード**を使用して、Windows Mobile デバイスに直接コピーできるデータベースを作成する
- dbinit ユーティリティを使用して、Windows Mobile デバイスに手動でコピーできるデータベースを作成する
- Interactive SQL で CREATE DATABASE 文を使用して、Windows Mobile デバイスに手動でコピーできるデータベースを作成する

**注意**
**データベース作成ウィザード**を使用して新しいデータベースを Windows Mobile デバイスに自動的にコピーする場合は、Sybase Central を ActiveSync または Windows Mobile Device Center と同じビット数にします (32 ビットまたは 64 ビット)。Windows XP 以前では、32 ビットのソフトウェアのみサポートされます。

Windows Mobile データベースを作成するときに決定することが必要な事項については、次の項を参照してください。


- 「Windows Mobile のトランザクションログ」417 ページ
- 「インストール時の考慮事項：Windows Mobile での ICU の使用」404 ページ
- 「インストール時の考慮事項：Windows Mobile での .NET Compact Framework の使用」405 ページ


**注意**
Windows Mobile 上でデータベースを実行すると、データベースサーバーで書き込みチェックサムが自動的に有効になります。この動作によって、データベースファイルの破損を速やかに検出できます。「チェックサムを使用した破損の検出」1120 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## Sybase Central を使用した Windows Mobile データベースの作成

**♦ Sybase Central で Windows Mobile のデータベースを作成して、それを Windows Mobile デバイスに直接コピーするには、次の手順に従います。**

Sybase Central には、Windows Mobile 用に簡単にデータベースを作成する機能があります。Sybase Central は、Windows Mobile のデータベースに必要な条件を満たし、生成されたデータベースファイルをデバイスにコピーするオプションを提供します。

1. Windows Mobile デバイスをデスクトップコンピューターに接続します。
2. デスクトップコンピューターで Sybase Central を起動します。32 ビットバージョンの Windows Mobile Device Center (ActiveSync) を使用する場合は、32 ビットバージョンの Sybase Central を起動します。Sybase Central は、Windows Mobile Device Center (ActiveSync) を使用してデータベースを Windows Mobile デバイスに転送します。
3. **[ツール]** » **[SQL Anywhere 12]** » **[データベースの作成]** をクリックします。
4. **[このコンピューターにデータベースを作成]** をクリックします。**[次へ]** をクリックします。
5. データベースファイルを格納するデスクトップコンピューターのフォルダーとファイル名を指定し、**[次へ]** をクリックします。
6. **[Windows Mobile にこのデータベースを作成]** をクリックし、**[次へ]** をクリックします。
7. **[データベースを Windows Mobile デバイスにコピー]** を選択し、**[次へ]** をクリックします。
8. データベースファイルをコピーする Windows Mobile のディレクトリを指定します。デフォルトロケーションは、メインデバイスのディレクトリです。

   **ヒント**
   Windows Mobile デバイスの *My Documents* ディレクトリにデータベースをコピーすると、データベースの開始が容易になります。

   Windows Mobile デバイスで **[サーバー起動オプション]** ウィンドウを使用してデータベースを開始するときに、**[参照]** ボタンを使用すると *My Documents* ディレクトリ内のデータベースファイルのみ検索できます。

   データベースを *My Documents* ディレクトリに保存していない場合は、**[サーバー起動オプション]** ウィンドウの **[データベース]** フィールドにデータベースのパスを入力する必要があります。

   オプションとして、**[コピー後にデスクトップのデータベースを削除]** を選択できます。

   デスクトップコンピューターのコピーを削除しない場合は、手順 5 で指定したディレクトリにデータベースファイルのコピーが格納されます。**[次へ]** をクリックします。
9. トランザクションログファイルを保存するディレクトリを指定します。**[次へ]** をクリックします。

Windows Mobile デバイスでは、ネットワークデータベースサーバーで初めてデータベースを開始したときに、データベースファイルと同じディレクトリにトランザクションログファイルが作成されます。

10. トランザクションログミラーを使用するかどうかを指定します。**[次へ]** をクリックします。

11. **[jConnect メタデータサポートをインストール]** オプションをクリアして、**[次へ]** をクリックします。

12. 適切なオプションを選択して、データベースの暗号化レベルを設定し、**[次へ]** をクリックします。

    強力な暗号化を選択した場合は、暗号化キーを指定する必要があります。キーには最低でも 16 文字の値を選択し、大文字と小文字、数字、文字、特殊文字を組み合わせて使用することをおすすめします。

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

13. ページサイズを選択し、**[次へ]** をクリックします。

14. **[追加設定の指定]** ページで、**[各データベースページでチェックサムを含む]** をクリックし、**[次へ]** をクリックします。

15. ウィザードの残りの指示に従い **[完了]** をクリックすると、データベースが作成されてデバイスにコピーされます。

    **注意**
    NCHAR データの照合順を指定します。NCHAR UCA ソートが必要ない場合、NCHAR 照合順は UTF8BIN にします。この場合、ICU ライブラリ (*dbicu12.dll* と *dbicudt12.dll*) はデータベースサーバーに必要ありません。**[提供されている次の照合を使用]** をクリックし、**UTF8BIN** を選択します。

16. Windows Mobile デバイスにコピーしているファイルの進行状況を示すウィンドウが表示されます。**[閉じる]** をクリックします。

17. データベースが Windows Mobile デバイスにコピーされたら、ファイルの場所を確認します。

    **[スタート]** » **[プログラム]** » **[ファイル エクスプローラー]** の順にタップして、データベースをコピーした Windows Mobile ディレクトリに移動します。

    データベースファイルが表示されます。トランザクションログファイルは、Windows Mobile デバイスで初めてデータベースを開始するときまで表示されません。

## dbinit を使用した Windows Mobile データベースの作成

**♦ dbinit ユーティリティを使用してデータベースを作成するには、次の手順に従います。**

Windows Mobile 用のデータベースを作成するために、dbinit ユーティリティを使用できます。ただし、このユーティリティから Windows Mobile デバイスに直接データベースをコピーすることはできません。dbinit ユーティリティを使用して作成したデータベースは、Windows Mobile デバイスに手動でコピーする必要があります。

1. コマンドプロンプトで、データベースを作成するディレクトリに移動します。次に例を示します。

   ```
   cd temp
   ```

2. 次のコマンドを実行して、データベースを作成します。

   ```
   dbinit -s database-name.db
   ```

   -s オプションは、データベースのチェックサムを有効にします。

   > **ヒント**
   > 暗号化やページサイズなど、データベースのプロパティも dbinit ユーティリティを使用して設定できます。「初期化ユーティリティ (dbinit)」958 ページを参照してください。

3. Windows Mobile デバイスにデータベースをコピーします。

   Windows Mobile デバイスへのデータベースのコピーについては、「Windows Mobile デバイスへのデータベースのコピー」422 ページを参照してください。

## CREATE DATABASE 文を使用した Windows Mobile データベースの作成

**♦ CREATE DATABASE 文を使用した Windows Mobile データベースの作成**

CREATE DATABASE 文を使用して、Interactive SQL でデスクトップコンピューターにデータベースを作成できます。ただし、このアプリケーションから Windows Mobile デバイスに直接データベースをコピーすることはできません。Windows Mobile デバイスにデータベースを手動でコピーする必要があります。

1. **[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[Interactive SQL]** をクリックします。

   **[接続]** ウィンドウが自動的に表示されない場合は、**[SQL]** » **[接続]** をクリックしてください。

2. **[ID]** タブをクリックし、次のフィールドに入力します。
   - **[ユーザー ID]**　**DBA** と入力します。
   - **[パスワード]**　**sql** と入力します。
   - **[アクション]**　**[ODBC データソースを使用した接続]** をクリックします。

- **[ODBC データソース名]**　**[参照]** をクリックし、「Windows Mobile デバイスに接続するための ODBC データソースの作成」412 ページで作成した **MobileServer** データソースを選択します。

3. Interactive SQL の **[SQL 文]** ウィンドウ枠で、次の文を入力します。

```
CREATE DATABASE 'c:¥¥temp¥¥database-name.db'
TRANSACTION LOG ON
CHECKSUM ON;
```

> **ヒント**
> 暗号化やページサイズなど、データベースのプロパティも CREATE DATABASE 文を使用して設定できます。「CREATE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

4. **[SQL]** メニューから、**[実行]** をクリックします。

   デスクトップコンピューターの *c:¥temp* ディレクトリに、データベースとトランザクションログが作成されます。

5. Windows Mobile デバイスにデータベースをコピーします。

   Windows Mobile デバイスへのデータベースのコピーについては、「Windows Mobile デバイスへのデータベースのコピー」422 ページを参照してください。

## Windows Mobile デバイスへのデータベースのコピー

### ♦ Windows Mobile デバイスへのデータベースのコピー

この項で説明する方法を使用すると、SQL Anywhere の既存のデータベースを Windows Mobile デバイスにコピーできます。ただし、Windows Mobile デバイスにデータベースをコピーした場合、Windows Mobile でサポートされていないデータベース機能は動作しないことに注意してください。

1. Windows Mobile デバイスをデスクトップコンピューターに接続します。
2. デスクトップコンピューターで Windows エクスプローラーを開きます。
3. コピーするデータベースが入っているディレクトリまで移動します。
4. データベースファイルを右クリックして、**[コピー]** をクリックします。
5. データベースファイルを保存する Windows Mobile デバイスのディレクトリまで移動します。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

> **ヒント**
> Windows Mobile デバイスで **[サーバー起動オプション]** ウィンドウを使用してデータベースを開始するときに、**[参照]** ボタンを使用すると *My Documents* フォルダーとそのサブフォルダー内のデータベースファイルのみ検索できます。
>
> データベースを *My Documents* ディレクトリに保存していない場合は、**[サーバー起動オプション]** ウィンドウの **[データベース]** フィールドにデータベースのパスを入力する必要があります。

6. Windows Mobile デバイスの [Windows エクスプローラー] ウィンドウで空白領域を右クリックして、**[貼り付け]** をクリックします。

   Windows Mobile デバイスにファイルがコピーされます。

**参照**

- 「Windows Mobile での SQL Anywhere 機能のサポート」434 ページ

## Windows Mobile でのデータベースの再構築

Windows Mobile のデータベースを再構築するには、以下のオプションがあります。

- 別のプラットフォームで Windows Mobile のデータベースを再構築し、Windows Mobile デバイスにデータベースをコピーする。この方法をおすすめします。
- dbmlsync を使用して空のデータベースを再配置する。
- dbremote を使用して空のデータベースを再配置する。
- Windows Mobile デバイス上で dbunload を使用する。このオプションはスマートフォンでは使用できません。

最初の 3 つのオプションは、Windows Mobile データベースをアップグレードするときに推奨されます。これらのオプションを利用できない場合は、Windows Mobile 上で dbunload を使用できます。Windows Mobile 上で dbunload を使用することを決定する前に、それに伴う次のような影響を検討する必要があります。

- データベースサーバーのテンポラリファイルのサイズ (アンロードや再ロードを実行すると、ファイルのサイズが数メガバイトになることがあります)
- dbunload および関連コンポーネント用に必要となる、追加の領域
- Windows Mobile デバイスのデータベースを複数コピーすることによって発生する、追加のコスト

Windows Mobile デバイスで dbunload を実行すると、一部のデバイスで利用可能なリソースよりも多くのリソースが必要になることがあるため、可能であれば、異なるプラットフォーム上でデータベースをアップグレードすることをおすすめします。

**注意**
Windows Mobile デバイス上で dbunload を実行する場合には、**SQL Anywhere 12 for Windows Mobile** の配備ウィザードの **[アンロード／再ロードのサポート]** オプションを選択する必要があります。SQL Anywhere for Windows Mobile を最初にインストールしたときにこのオプションを選択していない場合は、SQL Anywhere インストールを変更して、このサポートを追加できます。

### Windows Mobile で dbunload を使用する場合の注意

Windows Mobile デバイスで dbunload を使用するには、以下のタスクを実行しておく必要があります。

- 以下のファイルを、SQL Anywhere インストールディレクトリに配備します (デフォルトでは、*¥Program Files¥SQLAny12*)。
  - *dbsrv12.exe*
  - *dbsrv12.lic*
  - *dbunlspt.exe*
  - *dbunload.exe*
  - *dbrunsql.exe*
- 古い (バージョン 9 以前) データベースをアンロードし、*dbunlspt.exe* を含める場合は、*¥Windows* ディレクトリに次のファイルを展開する必要があります。
  - *scripts¥optdeflt.sql*
  - *scripts¥opttemp.sql*
  - *scripts¥unloadold.sql*
- 以下のファイルを *¥Windows* ディレクトリに配備します。
  - *dblgen12.dll*
  - *dblib12.dll*
  - *dbscript12.dll*
  - *dbtool12.dll*
  - *dbusen.dll*
- **Location** の文字列値 (REG_SZ) を、SQL Anywhere ソフトウェアディレクトリのレジストリキー *HKEY_LOCAL_MACHINE¥SOFTWARE¥Sybase¥SQL Anywhere¥12.0¥* 内に設定します。

## Windows Mobile でのデータベースの再構築

**♦ Windows Mobile でデータベースをアンロードするには、次の手順に従います (dbunload)。**

以下の手順をサードパーティ製 Windows Mobile アプリケーションに組み込んで、エンドユーザーに対する処理を自動化するようにできます。これを行うように選択する場合は、-qc と -q のいずれかまたは両方のオプションを指定して dbunload および dbrunsql を使用すること、または *dbtool12.dll* で DBUnload 関数を呼び出すことを検討してください。

1. Windows Mobile 以外のプラットフォームで、新しい空の SQL Anywhere 12 データベースを作成します。

CHAR 照合順は、既存のデータベースと一致する必要があります。NCHAR UCA ソートが必要ない場合、NCHAR 照合順は UTF8BIN にします。この場合、ICU ライブラリ (*dbicu12.dll* と *dbicudt12.dll*) はデータベースサーバーに必要ありません。

2. SQL Anywhere 12 ソフトウェアと空の SQL Anywhere データベースファイルを、Windows Mobile デバイスにコピーします。「Windows Mobile で dbunload を使用する場合の注意」424 ページを参照してください。

3. デバイス上で実行されているデータベースサーバーがないことを確認します。

4. 次のコマンドを実行します。

```
dbunload-path¥dbunload -c "UID=DBA;PWD=DBA-password;CHARSET=none;DBF=existing-database" unload-directory
```

5. dbunload が成功したことを確認したら、dbunload ウィンドウを閉じます。

6. 次のコマンドを実行します。

```
dbrunsql-path¥dbrunsql -c "UID=DBA;PWD=sql;CHARSET=none;DBF=new-empty-SQLAnywhere12database-file" -g- ¥reload.sql
```

7. dbrunsql が成功したことを確認したら、dbrunsql ウィンドウを閉じます。

8. Windows Mobile デバイスから *reload.sql* ファイルと *unload-directory* を削除します。

## Windows Mobile データベースのバックアップ

データが破損したりメディアに障害が発生した場合にデータを失わないように、バックアップとリカバリが重要になります。デバイスの盗難や紛失、またはメディアの障害によるデータ損失を防ぐには、Windows Mobile のデータベースを物理的に別の場所にバックアップするのが最適です。

バックアップとリカバリのユーティリティのほとんどは、Windows Mobile で使用できます。ただし、Windows Mobile ではユーティリティを使用してバックアップを物理的に別の場所に保存できないため、これらのユーティリティは役に立ちません (例外として、メインメモリのデータベースをリムーバブル SD カードにバックアップできます)。代わりに、データベースファイル全体をデスクトップコンピューターにコピーしてデータをバックアップします。また、同期を使用して Windows Mobile データベースの最新コピーをデスクトップコンピューターに維持することもできます。

**参照**

- 「Mobile Link 同期」『Mobile Link クイックスタート』

## Windows Mobile データベースの消去

**♦ デバイスのインターフェイスを使用してデータベースを消去するには、次の手順に従います。**

SQL Anywhere for Windows Mobile は、**データベース消去ウィザード**、DROP DATABASE 文、消去ユーティリティ (dberase) をサポートしていません。Windows Mobile デバイスからデータベースを手動で消去する必要があります。

実行中のデータベースは消去できません。

1. **[スタート]** » **[プログラム]** » **[ファイル エクスプローラー]** の順にタップして、削除するデータベースファイルのあるディレクトリに移動します。
2. データベースファイルをタップして押したままにします。
3. **[削除]** をタップします。
4. **[はい]** をタップして、削除を確認します。
5. トランザクションログファイルがある場合は、それを削除します。

**♦ Windows エクスプローラーを使用してデータベースを消去するには、次の手順に従います。**

SQL Anywhere for Windows Mobile は、**データベース消去ウィザード**、DROP DATABASE 文、消去ユーティリティ (dberase) をサポートしていません。Windows Mobile デバイスからデータベースを手動で消去する必要があります。

実行中のデータベースは消去できません。

1. Windows Mobile デバイスをクレードルに置き、Windows Mobile Device Center (ActiveSync) を介してデスクトップコンピューターに接続していることを確認します。
2. デスクトップコンピューターで Windows エクスプローラーを開きます。
3. データベースファイルが格納されている Windows Mobile ディレクトリまで参照します。
4. データベースファイルを右クリックして、**[削除]** をクリックします。
5. **[はい]** をクリックします。
6. トランザクションログファイルがある場合は、それを削除します。

## Windows Mobile のデータベースサーバー

Windows Mobile データベースサーバーは、明示的な要求がないかぎり、TCP/IP ネットワークリンクを開始しません。

Windows Mobile では、すでに最初の SQL Anywhere データベースサーバーを実行中に 2 番目のサーバーを起動しようとしても、最初のデータベースサーバーだけが動作します。Windows

Mobile アプリケーションではこれが標準の動作になります。そのため、Windows Mobile デバイスでは 2 つのデータベースサーバーを同時に実行することはできません。ただし、SQL Anywhere は 1 つのデータベースサーバー上で複数のデータベースを実行することはサポートします。

**参照**

- 「チュートリアル：Sybase Central から Windows Mobile データベースを実行する」427 ページ

## Windows Mobile のサーバーオプション

SQL Anywhere の動作とパフォーマンスをチューニングするために、データベースサーバーを起動するときにサーバーとデータベースのオプションを指定できます。さまざまなオプションを選択して、キャッシュで使用できるメモリ、データベースサーバー上のデータベースを起動するのに必要なパーミッションのレベル、使用するネットワークプロトコルなどの機能を指定できます。

Windows Mobile では、オプションは **[サーバー起動オプション]** ウィンドウで指定します。データベースサーバーのオプションをコマンドラインで設定する他の Windows オペレーティングシステムとは、この点で異なります。ほとんどのサーバーオプションは、Windows Mobile でも使用できます。

**参照**

- 「SQL Anywhere データベースサーバーの構文」189 ページ
- 「Windows Mobile でのデータベースサーバーオプションのサポート」437 ページ

# Windows Mobile の管理ユーティリティ

この項では、Windows Mobile データベースで SQL Anywhere データベース管理ユーティリティを使用するときに考慮すべき点を説明します。

## チュートリアル：Sybase Central から Windows Mobile データベースを実行する

Sybase Central は、SQL Anywhere を管理するためのグラフィカルユーザーインターフェイスを提供するデータベース管理ツールです。Sybase Central は、Mobile Link 同期など他の製品を管理する場合にも使用できます。

このチュートリアルを完了すると、サーバーの開始と停止、データベースサーバー上での単一または複数のデータベースの実行、データベースへの接続など、データベースサーバーに関連する主要タスクを実行できるようになります。

**始める前に**

チュートリアルで使用する Windows Mobile のデータベースを 2 つ作成する必要があります。

**♦ Windows Mobile デバイス用のデータベースを作成するには、次の手順に従います (Sybase Central の場合)。**

1. Windows Mobile デバイスをデスクトップコンピューターに接続します。
2. **[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[Sybase Central]** をクリックします。
3. **[ツール]** » **[SQL Anywhere 12]** » **[データベースの作成]** をクリックします。
4. **データベース作成ウィザード**の指示に従います。
5. **[データベースファイルの指定]** ページで **[参照]** をクリックし、データベースファイルのロケーションを選択します。データベースファイルに *Alpha* という名前を付けます。**[保存]** をクリックします。
6. **[Windows Mobile に作成]** ページで、**[Windows Mobile にこのデータベースを作成]** をクリックし、**[次へ]** をクリックします。

   Windows Mobile デバイスとの接続がテストされます。
7. **[データベースのコピー]** ページで **[データベースを Windows Mobile デバイスにコピー]** をクリックします。
8. **[Windows Mobile ファイル名]** フィールドに *¥My Documents¥Alpha.db* と入力します。
9. **[コピー後にデスクトップのデータベースを削除]** をクリックし、**[次へ]** をクリックします。
10. **[NCHAR データの照合順の指定]** ページで、**[提供されている次の照合を使用]** をクリックし、次に **[UTF8BIN]** をクリックします。

    詳細については、「インストール時の考慮事項 : Windows Mobile での ICU の使用」404 ページを参照してください。
11. **[完了]** をクリックします。
12. **[閉じる]** をクリックします。
13. この手順を繰り返して *¥My Documents¥Beta.db* というデータベースを作成します。
14. 「レッスン 1 : データベースサーバーの起動」428 ページに進みます。

## レッスン 1 : データベースサーバーの起動

この項では、Windows Mobile で単一のデータベースを実行するという簡単な例を紹介します。

**♦ サーバーでデータベースを開始するには、次の手順に従います。**

1. **[スタート]** » **[プログラム]** » **[SQLAny12]** の順にタップします。
2. **[Server]** をタップします。

3. **[データベース]** フィールドに開始するデータベースファイルの名前を入力するか、**[参照]** をタップして *My Documents* ディレクトリにある *Alpha.db* ファイルを指定します。

4. **[サーバー名]** フィールドに **MobileServer** と入力します。

5. **[オプション]** フィールドに **-gd all -x tcpip(port=2639)** と入力します。

   -gd オプションでパーミッションを設定すると、どのユーザーでもネットワークデータベースサーバーで追加のデータベースを開始できます。これは、レッスンの後半で必要になります。「-gd dbeng12/dbsrv12 サーバーオプション」233 ページを参照してください。

   データベースサーバーが使用するポートは 2639 です。ポート 2638 を使用することはおすすめしません。このポートは、デスクトップシステムで実行されている SQL Anywhere サーバーのデフォルト TCP/IP ポートでもあるためです。詳細については、「-x dbeng12/dbsrv12 サーバーオプション」289 ページを参照してください。

6. **[TCP/IP を使用]** をクリックします。

   Windows Mobile デバイスで実行中のデータベースにデスクトップコンピューターから接続するには、TCP/IP 接続が必要です。

7. **[OK]** をタップして、ネットワークデータベースサーバーで実行中のサンプルデータベースを開始します。

8. デバイスの **[Today]** 画面に移動します。

9. 画面の右下にあるデータベースサーバーアイコンをタップします。

   データベースサーバーメッセージウィンドウに「**要求を受信中です。**」というメッセージが表示されたら、次のレッスンに進むことができます。

10. 「レッスン 2：Windows Mobile データベースサーバーで複数のデータベースを開始する」429 ページに進みます。

## レッスン 2：Windows Mobile データベースサーバーで複数のデータベースを開始する

Windows Mobile デバイスでは、すでに最初の SQL Anywhere データベースサーバーを実行中に 2 番目のサーバーを起動しようとしても、最初のデータベースサーバーだけが動作します。Windows Mobile アプリケーションではこれが標準の動作になります。そのため、Windows Mobile デバイスでは 2 つのデータベースサーバーを同時に実行することはできません。複数のデータベースサーバーを実行する代わりに、1 つのサーバーで複数のデータベースを実行できます。

**♦ Sybase Central からデータベースに接続するには、次の手順に従います。**

1. **[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[Sybase Central]** をクリックします。

2. **[接続]** » **[SQL Anywhere 12 に接続]** をクリックします。
3. **[認証]** ドロップダウンリストで、**[データベース]** をクリックします。
   a. **[ユーザー ID]** フィールドに **DBA** と入力します。
   b. **[パスワード]** フィールドに **sql** と入力します。
4. **[アクション]** ドロップダウンリストから、**[別のコンピューターで稼働しているデータベースに接続]** をクリックします。
   a. **[ホスト]** フィールドに、**169.254.2.1** と入力します。
   b. **[ポート]** フィールドに、**2639** と入力します。
   c. **[サーバー名]** フィールドに **MobileServer** と入力します。
   d. **[データベース名]** フィールドに **Alpha** と入力します。
5. **[接続]** をクリックして、Windows Mobile デバイスで実行中の *Alpha.db* データベースに接続します。
6. データベースサーバーへの接続に失敗した場合は、「Windows Mobile デバイスに接続するための ODBC データソースの作成」412 ページを参照してください。

データベースサーバーを起動して Alpha データベースに接続したら、Windows Mobile デバイスで他のデータベースを開始できます。

**♦ ネットワークデータベースサーバーで 2 番目のデータベースを開始するには、次の手順に従います。**

1. Sybase Central の左ウィンドウ枠で、**[MobileServer]** を右クリックし、**[データベースの開始]** をクリックします。
2. **[データベースファイル]** フィールドに **¥My Documents¥Beta.db** と入力します。
3. **[OK]** をクリックして、ネットワークデータベースサーバーでデータベースを開始します。

ネットワークデータベースサーバーにデータベースがロードされます。デスクトップコンピューターから接続を開始します。

**♦ 2 番目のデータベースに接続するには、次の手順に従います。**

1. Sybase Central の左ウィンドウ枠で、**[Beta]** を右クリックし、**[接続]** をクリックします。
2. **[認証]** ドロップダウンリストから、**[データベース]** をクリックします。
   a. **[ユーザー ID]** フィールドに **DBA** と入力します。
   b. **[パスワード]** フィールドに **sql** と入力します。
3. **[アクション]** ドロップダウンリストから、**[別のコンピューターで稼働しているデータベースに接続]** をクリックします。
   a. **[ホスト]** フィールドに、**169.254.2.1** と入力します。

b. **[ポート]** フィールドに、**2639** と入力します。

c. **[サーバー名]** フィールドに **MobileServer** と入力します。

d. **[データベース名]** フィールドに **Beta** と入力します。

4. **[接続]** をクリックして、Windows Mobile デバイスで実行中の Beta データベースに接続します。

これで、Sybase Central を使って Alpha データベースと Beta データベースを表示して操作できます。

「レッスン 3：Windows Mobile でデータベースサーバーを停止する」431 ページに進みます。

## レッスン 3：Windows Mobile でデータベースサーバーを停止する

Windows Mobile デバイスでネットワークデータベースサーバーを停止する前に、デスクトップコンピューターから接続を切断する必要があります。

**♦ Windows Mobile データベースから接続を切断するには、次の手順に従います。**

1. Sybase Central で、**[接続]** » **[切断]** をクリックします。

2. Alpha データベースへの接続を選択します。

3. **[OK]** をクリックします。

4. **[接続]** » **[切断]** をクリックします。

   Alpha データベースが切断されます。

5. Beta データベースに対して手順 1 ～ 4 を繰り返します。

Sybase Central で Windows Mobile データベースから切断したので、ネットワークデータベースサーバーを停止できます。

**♦ サーバーを停止するには、次の手順に従います。**

1. Windows Mobile デバイスで **[Today]** 画面の右下にあるデータベースサーバーアイコンをタップします。

2. **[メニュー]** » **[シャットダウン]** をタップします。

**参考資料**

Sybase Central からデータベースに接続したら、データベース内のテーブルへのデータ追加、データベースオブジェクトの追加や編集、その他の管理タスクの実行、などを行うことができます。

Sybase Central からデータベースを管理することの詳細については、次の項を参照してください。


- 「SQL Anywhere 12 プラグイン」830 ページ
- 「テーブル、ビュー、インデックス」『SQL Anywhere サーバー SQL の使用法』


# チュートリアル：Interactive SQL を使用した Windows Mobile データベースの管理

Interactive SQL は、データベース内のデータの変更や問い合わせと、データベース構造の修正ができるアプリケーションです。Interactive SQL では、SQL 文を入力するためのウィンドウ枠が表示されます。また、クエリの進捗情報や結果セットを表示するウィンドウ枠も表示されます。

このチュートリアルでは、デスクトップコンピューターから Interactive SQL を使用して Windows Mobile デバイス上のデータベースを管理する方法を簡単に紹介します。まず、Interactive SQL から Windows Mobile デバイスのサンプルデータベースに接続する方法を学習します。接続したら、Interactive SQL を使用して SQL 文を実行できます。

## レッスン 1：サンプルデータベースの開始

Interactive SQL からサンプルデータベースに接続する前に、Windows Mobile デバイスでサンプルデータベースが実行されている必要があります。

**♦ サンプルアプリケーションを起動するには、次の手順に従います。**

1. **[スタート]** » **[プログラム]** » **[SQLAny12]** の順にタップします。
2. **[Server]** をタップします。
3. **[データベース]** フィールドに、サンプルデータベースのパスを入力します。デフォルトのロケーションは *¥My Documents¥demo.db* です。または、**[参照]** を使用して *demo.db* ファイルを見つけます。
4. **[サーバー名]** フィールドに **MobileServer** と入力します。
5. **[キャッシュ]** フィールドに **5MB** と入力します。

   Windows Mobile のデフォルトのキャッシュサイズは 600 KB ですが、パフォーマンスを向上させるために、これより大きいキャッシュサイズをおすすめします。
6. **[オプション]** フィールドに **-x tcpip(port=2639)** と入力します。

   データベースサーバーが使用するポートは 2639 です。ポート 2638 を使用することはおすすめしません。このポートは、デスクトップシステムで実行されている SQL Anywhere サーバーのデフォルト TCP/IP ポートでもあるためです。詳細については、「-x dbeng12/dbsrv12 サーバーオプション」289 ページを参照してください。
7. **[TCP/IP を使用]** をクリックします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Windows Mobile デバイスで実行中のデータベースにデスクトップコンピューターから接続するには、TCP/IP 接続が必要です。

8. **[OK]** をタップして、ネットワークデータベースサーバーで実行中のサンプルデータベースを開始します。

9. デバイスの **[Today]** 画面に移動します。

10. 画面の右下にあるデータベースサーバーアイコンをタップします。

    データベースサーバーメッセージウィンドウに「**要求を受信中です。**」というメッセージが表示されたら、次のレッスンに進むことができます。

11. 「レッスン 2 ：Interactive SQL を開始して接続する」433 ページに進みます。

## レッスン 2 ：Interactive SQL を開始して接続する

サンプルデータベースが Windows Mobile デバイスで実行されているので、デスクトップコンピューターから Interactive SQL を使用して接続し、データベースを表示したり管理したりできます。

**♦ Interactive SQL から Windows Mobile デバイスのデータベースに接続するには、次の手順に従います。**

1. デスクトップコンピューターで、**[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[Interactive SQL]** をクリックします。

2. **[認証]** ドロップダウンリストから、**[データベース]** をクリックします。
   a. **[ユーザー ID]** フィールドに **DBA** と入力します。
   b. **[パスワード]** フィールドに **sql** と入力します。

3. **[アクション]** ドロップダウンリストから、**[別のコンピューターで稼働しているデータベースに接続]** をクリックします。
   a. **[ホスト]** フィールドに、**169.254.2.1** と入力します。
   b. **[ポート]** フィールドに、**2639** と入力します。
   c. **[サーバー名]** フィールドに **MobileServer** と入力します。
   d. **[データベース名]** フィールドに **Demo** と入力します。

4. **[接続]** をクリックして、Windows Mobile デバイスで実行中のサンプルデータベースに接続します。

   データベースサーバーへの接続に失敗した場合は、「Windows Mobile デバイスに接続するための ODBC データソースの作成」412 ページを参照してください。

5. 「レッスン 3：Windows Mobile データベースに対してクエリを実行する」434 ページに進みます。

## レッスン 3：Windows Mobile データベースに対してクエリを実行する

Interactive SQL の重要な使用目的の 1 つは、テーブルデータをブラウズすることです。Interactive SQL では、データベースサーバーに要求を送信することによって情報を検索します。一方、データベースサーバーは、情報を調べ、それを Interactive SQL に返します。

**♦ Windows Mobile データベースで SQL 文を実行するには、次の手順に従います。**

1. [SQL 文] ウィンドウ枠で、次の文を実行します。

   ```
   SELECT * FROM Employees;
   ```

2. **[SQL]** メニューから、**[実行]** をクリックして文を実行します。

   Employees テーブルのすべてのデータが、**[結果]** ウィンドウ枠に表示されます。

3. **[SQL]** メニューから、**[切断]** をクリックして Windows Mobile データベースから切断します。

### 参考資料

Interactive SQL からデータベースに接続すると、データの表示や操作、またデータベースオブジェクトの追加や変更を行うことができます。

### 参照


- 「Interactive SQL」840 ページ
- 「クエリ」『SQL Anywhere サーバー SQL の使用法』
- 「クエリ結果の要約、グループ化、ソート」『SQL Anywhere サーバー SQL の使用法』
- 「ジョイン：複数テーブルからのデータ検索」『SQL Anywhere サーバー SQL の使用法』
- 「SQL 文」『SQL Anywhere サーバー SQL リファレンス』


# Windows Mobile での SQL Anywhere 機能のサポート

この項では、SQL Anywhere のコンポーネントや機能のうち、Windows Mobile でサポートされていないか、動作が異なるものをリストします。サポートされていない機能の代わりになる機能がある場合は、それについても説明します。

Windows Mobile でサポートされるコンポーネントとサポートされないコンポーネントの詳細については、http://www.ianywhere.jp/tech/1061806-os_components.html を参照してください。

SQL Anywhere には、データベースを管理するための数々のツールが付属しています。たとえば、Sybase Central、Interactive SQL、コマンドラインユーティリティなどがあります。これらの管理ツールは、Windows Mobile では使用できません。データベースの管理は、Windows Mobile デバイスに接続している Windows ベースのデスクトップコンピューターから行います。

詳細については、「Windows Mobile の管理ユーティリティ」427 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| コンポーネントまたは機能 | 考慮事項 |
| --- | --- |
| アプリケーションプロファイリング | Windows Mobile で実行しているデータベースのトレーシングセッションを作成する場合は、**データベーストレーシングウィザード**を使用してトレースを設定する必要があります。**アプリケーションプロファイリングウィザード**は使用できません。また、Windows Mobile デバイスのデータをトレースし、デスクトップコンピューター上のデータベースサーバーで実行している Windows Mobile データベースのコピーに格納する必要があります。Windows Mobile デバイスからトレーシングデータベースを自動的に作成することはできません。また、Windows Mobile デバイス上のローカルデータベースにトレースすることはできません。 「アプリケーションプロファイリング」『SQL Anywhere サーバー SQL の使用法』を参照してください。 |
| データベースミラーリング | Windows Mobile ではサポートされていません。 |
| 外部ストアドプロシージャー | Windows Mobile ではサポートされていません。 |
| SQL Anywhere JDBC ドライバー | Windows Mobile ではサポートされていません。Windows Mobile 上で jConnect を使用できます。 |
| 外部環境 | Windows Mobile ではサポートされていません。 |
| jConnect | jConnect ドライバーは、Windows Mobile 用データベースを作成するときに有効にできます。これは、Java をサポートしているデスクトップコンピューターにデータベースを移動する場合に便利です。ただし、データベースサイズが大きくなり、jConnect 機能を使用しない場合でも、データベースを実行するための必要メモリが約 200 KB 増えます。Windows Mobile の限られたメモリ環境でデータベースを実行する場合は、この追加のメモリ要件を考慮してください。「Windows Mobile の jConnect」417 ページを参照してください。 |
| Kerberos 認証 | Windows Mobile ではサポートされていません。 |
| LDAP | Windows Mobile ではサポートされていません。 |

| コンポーネントまたは機能 | 考慮事項 |
|---|---|
| ODBC クライアント | Windows Mobile では、ODBC ドライバーマネージャーまたは ODBC アドミニストレーターを提供していないため、SQL Anywhere はファイルに保管されている ODBC データソースを使用します。「Windows Mobile での ODBC データソース」47 ページを参照してください。 |
| Sybase Open Client | Windows Mobile ではサポートされていません。 |
| 並列バックアップ | Windows Mobile ではサポートされていません。 |
| パーソナルデータベースサーバー (dbeng12) | Windows Mobile でサポートされているのはネットワークデータベースサーバー (*dbsrv12.exe*) のみです。この実行プログラムは、ローカル接続とネットワーク経由のクライアント／サーバー通信をサポートしています。 |
| リモートデータアクセス (ディレクトリアクセスサーバーを含む) | Windows Mobile ではサポートされていません。 |

### Windows Mobile での SQL 文のサポート

この項では、Windows Mobile でサポートされていないか、機能が異なったり機能に制限のある SQL 文について説明します。

SQL 文の全リストについては、「SQL 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

| SQL 文 | 考慮事項 |
|---|---|
| BACKUP 文 | Windows Mobile でサポートされているのは、BACKUP DATABASE DIRECTORY 句のみです。 |
| CREATE DATABASE 文 | CREATE DATABASE 文を使用してデスクトップコンピューターのデータベースを初期化し、これを後で Windows Mobile デバイスにコピーできます。「Windows Mobile データベースサーバーの作成」418 ページを参照してください。 |
| CREATE EVENT 文 | DiskSpace イベントタイプは、Windows Mobile ではサポートされていません。ただし、この文は GlobalAutoincrement または ServerIdle イベントタイプの定義に使用できます。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| SQL 文 | 考慮事項 |
|---|---|
| CREATE EXISTING TABLE 文 | Windows Mobile ではサポートされていません。 |
| CREATE EXTERNLOGIN 文 | Windows Mobile ではサポートされていません。 |
| CREATE FUNCTION 文 | データベースで使用するユーザー定義の SQL 関数を作成する場合は、Windows Mobile でも CREATE FUNCTION 文を使うことができます。Windows Mobile では EXTERNAL NAME 句はサポートされていません。 |
| CREATE SERVER 文 | Windows Mobile ではサポートされていません。 |
| CREATE TABLE 文 | プロキシテーブルを作成するための CREATE TABLE 文の AT 句は、Windows Mobile ではサポートされていません。 |
| DROP DATABASE 文 | Windows Mobile ではサポートされていません。 |
| DROP SERVER 文 | Windows Mobile ではサポートされていません。 |
| INSTALL JAVA 文 | Windows Mobile ではサポートされていません。 |
| REMOVE JAVA 文 | Windows Mobile ではサポートされていません。 |
| REORGANIZE TABLE 文 | Windows Mobile ではサポートされていません。 |
| RESTORE DATABASE 文 | Windows Mobile ではサポートされていません。 |
| START JAVA 文 | Windows Mobile ではサポートされていません。 |
| STOP JAVA 文 | Windows Mobile ではサポートされていません。 |

### Windows Mobile でのデータベースサーバーオプションのサポート

この項では、Windows Mobile でサポートされていないか、機能が異なるデータベースサーバーオプションについて説明します。

| オプション | 考慮事項 |
| --- | --- |
| *@data* オプション | Windows Mobile ではサポートされていません。 |
| -? サーバーオプション | Windows Mobile ではサポートされていません。 |
| -cm サーバーオプション | Windows Mobile ではサポートされていません。 |
| -cw サーバーオプション | Windows Mobile ではサポートされていません。 |
| -ec サーバーオプション | 強力な通信暗号化 (TLS) は、Windows Mobile ではサポートされていません。**none** (なし) と **simple** (単純) の設定だけがサポートされています。 「-ec dbeng12/dbsrv12 サーバーオプション」223 ページを参照してください。 |
| -gb サーバーオプション | Windows Mobile ではサポートされていません。 |
| -ge サーバーオプション | Windows Mobile ではサポートされていません。 |
| -gna サーバーオプション | Windows Mobile ではサポートされていません。 |
| -gnh サーバーオプション | Windows Mobile ではサポートされていません。 |
| -gnl サーバーオプション | Windows Mobile ではサポートされていません。 |
| -gns サーバーオプション | Windows Mobile ではサポートされていません。 |
| -gt サーバーオプション | Windows Mobile ではサポートされていません。 |
| -gtc サーバーオプション | Windows Mobile ではサポートされていません。 |
| -qi サーバーオプション | 実行中、ネットワークデータベースサーバーは Windows Mobile デバイスの [Today] 画面の右下にアイコンとして表示されます。この機能は無効にできません。 |

| オプション | 考慮事項 |
| --- | --- |
| -s サーバーオプション | Windows Mobile ではサポートされていません。 |
| -tmf サーバーオプション | Windows Mobile ではサポートされていません。 |
| -tmt サーバーオプション | Windows Mobile ではサポートされていません。 |
| -u サーバーオプション | Windows Mobile ではサポートされていません。 |
| -ua サーバーオプション | Windows Mobile ではサポートされていません。 |
| -uc サーバーオプション | Windows Mobile ではサポートされていません。 |
| -ud サーバーオプション | Windows Mobile ではサポートされていません。 |
| -uf サーバーオプション | Windows Mobile ではサポートされていません。 |
| -ui サーバーオプション | Windows Mobile ではサポートされていません。 |
| -ut サーバーオプション | Windows Mobile ではサポートされていません。 |
| -ux サーバーオプション | Windows Mobile ではサポートされていません。 |
| -xp サーバーオプション | Windows Mobile ではサポートされていません。 |
| -ze サーバーオプション | Windows Mobile ではサポートされていません。 |

### Windows Mobile での Sybase Central ウィザードのサポート

次の表に、Windows Mobile でサポートされていない Sybase Central のウィザード、機能が異なるウィザード、その代わりとなる方法 (可能な場合) をリストします。

| ウィザード | 考慮事項 |
| --- | --- |
| データベースバックアップウィザード | アーカイブバックアップは、Windows Mobile ではサポートされていません。したがって、**データベースバックアップウィザード**はサポートされていません。「バックアップの種類」1072 ページを参照してください。<br><br>別の方法として、Windows Mobile では、**バックアップイメージ作成ウィザード**を使用できます。このウィザードは、データベースファイルとトランザクションログファイルのバックアップをそれぞれ別のファイルとして作成します。「データベースバックアップウィザードを使用した、アーカイブバックアップの作成」1083 ページを参照してください。 |
| ログファイル設定の変更ウィザード | Windows Mobile ではサポートされていません。 |
| データベース作成ウィザード | このウィザードは、Windows Mobile で使用するデータベースを作成するオプションを提供します。この場合、Sybase Central を実行しているデスクトップコンピューター上に Windows Mobile サービスがインストールされていることが前提となります。「Windows Mobile データベースサーバーの作成」418 ページを参照してください。 |
| メンテナンスプラン作成ウィザード | 以下のオプションは、Windows Mobile では利用できません。<br><br>● フルアーカイブバックアップ<br>● テープにバックアップ<br>● メンテナンスプランレポートを電子メールで送信する |
| データベース消去ウィザード | Windows Mobile ではサポートされていません。 |
| データベース移行ウィザード | Windows Mobile ではサポートされていません。 |
| データベースリストアウィザード | Windows Mobile ではサポートされていません。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| ウィザード | 考慮事項 |
| --- | --- |
| サービス作成ウィザード | Windows Mobile ではサポートされていません。 |
| ログファイル変換ウィザード | Windows Mobile ではサポートされていません。 |
| データベースアンロードウィザード | このウィザードは、データベースファイルが格納されている Windows Mobile ディレクトリにはマッピングできません。ただし、Windows Mobile データベースをデスクトップコンピューターにコピーすると、**データベースアンロードウィザード**を使用してアンロードできます。 |
| データベースアップグレードウィザード | このウィザードは、Windows Mobile ではサポートされていません。ただし、Windows Mobile データベースをデスクトップコンピューターにコピーし、このウィザードを使ってから Windows Mobile デバイスにコピーし直すと、アップグレードできます。「SQL Anywhere サーバーのアップグレード」『SQL Anywhere 12 変更点とアップグレード』を参照してください。 |

### Windows Mobile での SQL Remote のサポート

SQL Remote は、次の例外を除いて Windows Mobile でサポートされています。

| コンポーネントまたは機能 | 考慮事項 |
| --- | --- |
| 抽出ユーティリティ (dbxtract) | Windows Mobile は、このユーティリティをサポートしていません。必要な場合は、Windows Mobile データベースをデスクトップコンピューターにコピーすると、抽出ユーティリティを使えるようになります。 |

# データベース設定

この項では、SQL Anywhere に必要なファイル、データベースの制限事項、データベースプロパティとオプションの設定方法について説明します。ここでは、SQL Anywhere のインストール環境を設定して、国際言語の問題を処理する方法についても説明します。

## SQL Anywhere の環境変数

SQL Anywhere では、環境変数にさまざまな情報を登録しています。設定しなければならない環境変数は、状況によって異なります。

SQL Anywhere サーバーの場合、特定のサーバーに設定された環境変数を参照するには、サーバーを起動するときに -ze オプションを指定します。「-ze dbeng12/dbsrv12 オプション」297 ページを参照してください。

### Windows での環境変数の設定

SQL Anywhere のインストーラーによって、コンピューターのプロパティに環境変数 PATH と SQLANY12 が作成されます。既存する場合は、変更されます。SQL Anywhere をインストールしたら、コンピューターを再起動して、これらの環境変数を反映する必要があります。

その他の環境変数は、コンピューターのプロパティを変更して設定できます。また、SET コマンドを使用してコマンドプロンプトやバッチファイルから変更することも可能です。

## Unix および Mac OS X 環境変数

SQL Anywhere のインストール後、システムが SQL Anywhere アプリケーションの場所を特定して実行できるように、各ユーザーはシステムの環境変数を設定します。SQL Anywhere インストーラーは、ユーザーの環境変数に対応するために *sa_config.sh* と *sa_config.csh* という 2 つのファイルを作成します。これらのファイルは、*$SQLANY12/bin32* と *$SQLANY12/bin64* にインストールされます。各ファイルは、必要なすべてのユーザー環境変数を設定します。

名前のとおり、1 つのファイルは Bourne シェル (*sh*) とその派生シェル (*ksh* や *bash* など) で動作するように設計されています。もう 1 つのファイルは、C シェル (*csh*) とその派生シェル (*tcsh* など) で動作するように設計されています。

一部の文は、各バッチファイルでコメントアウトされます。システム管理者は、システムの設定に応じてこれらのファイルを編集し、コメントを削除できます。

SQL Anywhere アプリケーションは、いつくかの方法で実行できます。

1. *sa_config* ファイルに指定されている環境変数をシステム環境に追加した場合は、X-Window Server などの GUI から起動したり、ターミナルウィンドウでアプリケーション名を入力したりしてアプリケーションを実行できます。

2. *sa_config* ファイルのいずれかのソースを指定してある場合は、ターミナルウィンドウでアプリケーション名を入力することでアプリケーションを実行できます。「UNIX および Mac OS X ファイル」444 ページを参照してください。

3. *$SQLANY12/bin32s* と *$SQLANY12/bin64s* には、SQL Anywhere アプリケーションと同じ名前のスクリプトが含まれています。これらのスクリプトは、適切な環境変数を設定してから、アプリケーションを起動します。このため、対応するスクリプトを実行することで、アプリケーションを実行できます。スクリプトを実行する前に *sa_config* ファイルのソースを指定する必要はありません。

### Mac OS X の Finder の環境変数の設定

SQL Anywhere のインストーラーによって、環境変数 DYLD_LIBRARY_PATH、ODBCINI、PATH、SQLANY12 が設定されます。再起動する必要はありません。

ターミナルセッションは、Finder の環境変数を継承しません。ターミナルセッションの環境変数を設定する方法については、「UNIX および Mac OS X ファイル」444 ページを参照してください。

## UNIX および Mac OS X ファイル

ファイルの「ソースを指定する」とは、シェルの現在のインスタンスでテキストファイルに含まれるコマンドを実行することを指します。これは、シェルに組み込まれたコマンドを使用することで実行できます。

Bourne シェルとその派生シェルでは、このコマンド名は . (単一のピリオド) です。たとえば、SQL Anywhere が */opt/sqlanywhere12* にインストールされている場合、次の文を使用して *sa_config.sh* のソースを指定します。

```
. /opt/sqlanywhere12/bin32/sa_config.sh
```

Mac の場合は、次のコマンドを実行して *sa_config.sh* のソースを指定します。

```
. Applications/SQLAnywhere12/System/bin32/sa_config.sh
```

C シェルとその派生シェルの場合、コマンドは source です。たとえば、SQL Anywhere が */opt/sqlanywhere12* にインストールされている場合、次のコマンドを使用して *sa_config.csh* のソースを指定します。

```
source /opt/sqlanywhere12/bin32/sa_config.csh
```

### サンプルの設定

SQL Anywhere 12 のサンプルを使用する前に、サンプルで使用する SQL Anywhere のビット数に対応する設定スクリプトを実行してください。たとえば、32 ビットのソフトウェアでサンプルを使用できるように設定するには、次のように実行します。

```
. /opt/sqlanywhere12/samples/sample_config32.sh
```

Mac の場合、32 ビットのソフトウェアでサンプルを使用できるように設定するには、次のように実行します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

. Applications/SQLAnywhere12/samples/sample_config32.sh

# DYLD_LIBRARY_PATH 環境変数 [Mac OS X]

Mac OS X 上で SQL Anywhere アプリケーションが必要とするライブラリの実行時の検索場所になるディレクトリを指定します。

### 構文

**DYLD_LIBRARY_PATH=***path-list*

### デフォルト

*/Applications/SQLAnywhere12/System/lib32*

### 備考

インストーラーによって作成される *sa_config.sh* ファイルと *sa_config.csh* ファイルは、これを含むすべての環境変数を作成または変更するスクリプトです。

### 参照


- 「LD_LIBRARY_PATH 環境変数 [Linux と Solaris]」445 ページ
- 「LIBPATH 環境変数 [IBM AIX]」446 ページ
- 「SHLIB_PATH 環境変数 [HP-UX]」453 ページ
- 「Unix および Mac OS X 環境変数」443 ページ


# LD_LIBRARY_PATH 環境変数 [Linux と Solaris]

Linux と Solaris 上で SQL Anywhere アプリケーションが必要とするライブラリの実行時の検索場所になるディレクトリを指定します。

### 構文

**LD_LIBRARY_PATH=***path-list*

### デフォルト

- **Linux**
  - */opt/sqlanywhere12/lib32:/opt/sqlanywhere12/sun/jre160_x86/lib/i386/client:/opt/sqlanywhere12/sun/jre160_x86/lib/i386/server:/opt/sqlanywhere12/sun/jre160_x86/lib/i386:/opt/sqlanywhere12/sun/jre160_x86/lib/i386/native_threads* (32 ビット)
  - */opt/sqlanywhere12/lib64:/opt/sqlanywhere12/sun/jre160_x86/lib/i386/client:/opt/sqlanywhere12/sun/jre160_x86/lib/i386/server:/opt/sqlanywhere12/sun/jre160_x86/lib/i386:/opt/sqlanywhere12/sun/jre160_x86/lib/i386/native_threads* (64 ビット)
- **Solaris Intel**
  - */opt/sqlanywhere12/lib32* (32 ビット)
  - */opt/sqlanywhere12/lib64* (64 ビット)

- **Solaris SPARC**
  - */opt/sqlanywhere12/lib32:/opt/sqlanywhere12/sun/jre160/lib/sparc/client:/opt/sqlanywhere12/sun/jre160/lib/sparc/server:/opt/sqlanywhere12/sun/jre160/lib/sparc:/opt/sqlanywhere12/sun/jre160/lib/sparc/native_threads:${LD_LIBRARY_PATH:-}"* (32 ビット)
  - */opt/sqlanywhere12/lib64:"/opt/sqlanywhere12/sun/jre160/lib/sparc/client:/opt/sqlanywhere12/sun/jre160/lib/sparc/server:/opt/sqlanywhere12/sun/jre160/lib/sparc:/opt/sqlanywhere12/sun/jre160/lib/sparc/native_threads* (64 ビット)

### 備考

インストーラーによって作成される *sa_config.sh* ファイルと *sa_config.csh* ファイルは、これを含むすべての環境変数を作成または変更するスクリプトです。

### 参照


- 「DYLD_LIBRARY_PATH 環境変数 [Mac OS X]」445 ページ
- 「LIBPATH 環境変数 [IBM AIX]」446 ページ
- 「SHLIB_PATH 環境変数 [HP-UX]」453 ページ
- 「Unix および Mac OS X 環境変数」443 ページ


## LIBPATH 環境変数 [IBM AIX]

IBM AIX 上で SQL Anywhere アプリケーションが必要とするライブラリの実行時の検索場所になるディレクトリを指定します。

### 構文

**LIBPATH=***path-list*

### デフォルト

- */usr/lpp/sqlanywhere12/lib32* (32 ビットのプラットフォーム)
- */usr/lpp/sqlanywhere12/lib64* (64 ビットのプラットフォーム)

### 備考

インストーラーによって作成される *sa_config.sh* ファイルと *sa_config.csh* ファイルは、これを含むすべての環境変数を作成または変更するスクリプトです。

### 参照


- 「DYLD_LIBRARY_PATH 環境変数 [Mac OS X]」445 ページ
- 「LD_LIBRARY_PATH 環境変数 [Linux と Solaris]」445 ページ
- 「SHLIB_PATH 環境変数 [HP-UX]」453 ページ
- 「Unix および Mac OS X 環境変数」443 ページ


 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

# ODBCHOME 環境変数 [UNIX]

*.odbc.ini* ファイルのロケーションを指定します。

### 構文

**ODBCHOME=***odbc-ini-directory*

### 備考

*.odbc.ini* ファイルは、ODBC データソースを含むシステム情報ファイルです。このファイルの名前が *.odbc.ini* ではない場合は、ODBCINI または ODBC_INI 環境変数を使用してロケーションを指定する必要があります。

ODBC データソースの検索アルゴリズムの詳細については、「UNIX での ODBC データソース」48 ページを参照してください。

### 参照


- 「ODBCINI と ODBC_INI 環境変数 [UNIX]」447 ページ
- 「Unix および Mac OS X 環境変数」443 ページ


# ODBCINI と ODBC_INI 環境変数 [UNIX]

ODBC データソースを含むシステム情報ファイルのパスと名前を指定します。

### 構文

**ODBCINI=***odbc-ini-file*

**ODBC_INI=***odbc-ini-file*

### 備考

システム情報ファイルは、これらのいずれかの環境変数を使用されて指定している場合は、*.odbc.ini* という名前でなくてもかまいません。2 つの環境変数が用意されているのは、他製品との互換性を確保するためです。

ODBC データソースの検索アルゴリズムの詳細については、「UNIX での ODBC データソース」48 ページを参照してください。

### 参照


- 「ODBCHOME 環境変数 [UNIX]」447 ページ
- 「Unix および Mac OS X 環境変数」443 ページ


# PATH 環境変数

SQL Anywhere 実行プログラムがあるディレクトリのロケーションを指定します。

### 構文

**PATH=***path-list*

### デフォルト

次のパスは、対応するコンポーネントがインストールされた場合にのみ追加されます。

| オペレーティングシステム | デフォルトのロケーション |
|---|---|
| Windows (32 ビット) | *C:¥Program Files¥SQL Anywhere 12¥bin32* |
| Windows (64 ビット) | *C:¥Program Files¥SQL Anywhere 12¥bin64* |
| Mac OS X (32 ビット) | */Applications/SQLAnywhere12/System/bin32* |
| Mac OS X (64 ビット) | */Applications/SQLAnywhere12/System/bin64* |
| IBM AIX (32 ビット) | */usr/lpp/sqlanywhere12/bin32* |
| IBM AIX (64 ビット) | */usr/lpp/sqlanywhere12/bin64* |
| Solaris Sparc (32 ビットと 64 ビット) | */opt/sqlanywhere12/sun/jre160/bin* |
| Linux (32 ビット) | */opt/sqlanywhere12/bin32:/opt/sqlanywhere12/sun/jre160_x86/bin/* |
| Linux (64 ビット) | */opt/sqlanywhere12/bin32*:/opt/sqlanywhere12/sun/jre160_x64/bin |
| その他の UNIX オペレーティングシステム (32 ビット) | */opt/sqlanywhere12/bin32* |
| その他の UNIX オペレーティングシステム (64 ビット) | */opt/sqlanywhere12/bin64* |

### 備考

Windows では、PATH 環境変数の設定がインストーラーによって変更され、SQL Anywhere 実行プログラムの保存先ディレクトリが追加されます。

UNIX では、これを含むすべての環境変数を作成または変更するスクリプトとして、*sa_config.sh* ファイルと *sa_config.csh* ファイルがインストーラーによって作成されます。

### 参照


- 「Windows での環境変数の設定」443 ページ
- 「Unix および Mac OS X 環境変数」443 ページ
- 「UNIX および Mac OS X ファイル」444 ページ

# SACHARSET 環境変数

SQL Anywhere が使用する文字セットを指定します。

### 構文

**SACHARSET=***charset-label*

### 備考

*charset-label* は文字セットラベルです。

Windows と UNIX の各オペレーティングシステムに推奨される文字セットラベルの詳細については、「推奨文字セットと照合」506 ページを参照してください。

dbinit -le オプションを使用すると、SQL Anywhere データベースで使用可能なすべての文字セットラベルのリストを取得できます。dbinit -l オプションは、SQL Anywhere データベースで使用可能な照合のリストを返します。「初期化ユーティリティ (dbinit)」958 ページを参照してください。

SACHARSET を指定しなかった場合は、オペレーティングシステムの文字セットが使用されます。

### 参照


- 「Windows での環境変数の設定」443 ページ
- 「Unix および Mac OS X 環境変数」443 ページ


# SADIAGDIR 環境変数

SQL Anywhere 診断ディレクトリのロケーションを指定します。

### 構文

**SADIAGDIR=***diagnostic-information-directory*

### デフォルト

この環境変数はデフォルトで設定されていません。

### 備考

この環境変数は、SQL Anywhere 診断ディレクトリのデフォルトのロケーションに優先されます。SQL Anywhere プログラムは、ソフトウェアに致命的なエラーがある場合に、このディレクトリを使用してレポートを格納します。

SQL Anywhere では、クラッシュレポートと機能診断情報が診断ディレクトリに保存されます。クラッシュレポートが書き込まれる診断ディレクトリのロケーションは、SADIAGDIR 環境変数の設定によって決まります。

環境変数で指定されたディレクトリが存在しない場合、データベースサーバーは環境変数が設定されていないものとして処理を実行します。

### 参照


- 診断ディレクトリの場所1038 ページ
- 「サポートユーティリティ (dbsupport)」1027 ページ
- 「SQL Anywhere のエラーレポート」1210 ページ
- 「Windows での環境変数の設定」443 ページ
- 「Unix および Mac OS X 環境変数」443 ページ


## SALANG 環境変数

SQL Anywhere の言語コードを指定します。

### 構文

**SALANG=***language-code*

### 備考

*language-code* は言語を表す 2 文字のコードです。たとえば、**SALANG=DE** の場合、デフォルトの言語がドイツ語に設定されます。

サポートされている言語コードの詳細については、「ロケール言語」485 ページを参照してください。

次の方法で最初に値を返すものが、デフォルトの言語を特定します。

1. SALANG 環境変数をチェックします。
2. (Windows の場合) インストール時、または *dblang.exe* によって設定されたレジストリをチェックします。「言語選択ユーティリティ (dblang)」983 ページを参照してください。
3. オペレーティングシステムに対して言語情報を問い合わせます。
4. 言語情報が設定されていない場合は、英語がデフォルトになります。

### 参照


- 「言語選択ユーティリティ (dblang)」983 ページ
- 「インストール時のレジストリ設定」466 ページ
- 「Windows での環境変数の設定」443 ページ
- 「Unix および Mac OS X 環境変数」443 ページ


## SALOGDIR 環境変数

*backup.syb* ファイルのロケーションを指定します。

### 構文

**SALOGDIR=***directory-name*

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 備考

SALOGDIR 環境変数が設定されている場合、バックアップ履歴ファイル *backup.syb* を書き込むことができるディレクトリのパスが含まれることが想定されます。このファイルは BACKUP 文または RESTORE 文を実行するたびに更新されます。

Windows では、次のうち最初の書き込み可能なロケーションに *backup.syb* ファイルが書き込まれます。

1. SALOGDIR 環境変数
2. *%ALLUSERSPROFILE%¥SQL Anywhere 12*。
3. 現在のドライブのルートディレクトリ。
4. 現在のディレクトリ。

Windows CE では、次のうち最初の書き込み可能なロケーションに *backup.syb* ファイルが書き込まれます。

1. SALOGDIR 環境変数
2. データベースサーバーが起動されたディレクトリ。
3. ルートディレクトリ。
4. 現在のディレクトリ。

UNIX では、次のうち最初の書き込み可能なロケーションに *backup.syb* ファイルが書き込まれます。

1. SALOGDIR 環境変数
2. *$HOME/.sqlanywhere12*。
3. 現在のディレクトリ。

### 参照


- 「BACKUP 文」『SQL Anywhere サーバー SQL リファレンス』
- 「Windows での環境変数の設定」443 ページ
- 「Unix および Mac OS X 環境変数」443 ページ


## SATMP 環境変数

データベースサーバーとテンポラリディレクトリを必要とする SQL Anywhere コマンドラインユーティリティが使用するテンポラリファイルのロケーションを指定します。

### 構文

**SATMP=***directory-name*

**備考**

SQL Anywhere では、データベースサーバー関連のテンポラリファイル (すべてのプラットフォームで作成) と通信関連のテンポラリファイル (UNIX のクライアントとサーバーでのみ作成) の 2 種類のテンポラリファイルが作成されます。

SATMP 環境変数には、データベースサーバーとテンポラリディレクトリを必要とする SQL Anywhere コマンドラインユーティリティが使用するテンポラリファイルのロケーションが指定されます。このファイルは、データベースサーバーをサービスとして実行する場合に役立ちます。テンポラリファイルをほかのプログラムがアクセスできないディレクトリに保存できるからです。

データベースサーバーの起動時に -dt オプションでテンポラリファイルのロケーションを指定しなかった場合は、SATMP 環境変数の値に基づいてテンポラリファイルの保存場所が決定されます。SATMP 環境変数が設定されていない場合は、環境変数 TMP、TMPDIR、TEMP のうち、最初に見つかった変数の値が使用されます。Windows では、これらの環境変数がいずれも存在しない場合、現在のディレクトリが使用されます。UNIX では、これらの環境変数のいずれも存在しない場合、*/tmp* が使用されます。

Windows Mobile では、サーバーのテンポラリフォルダーとして使用するディレクトリをレジストリに指定できます。 Windows Mobile 上のテンポラリファイルのロケーションについては、「Windows Mobile でのレジストリ設定」466 ページを参照してください。

UNIX では、共有メモリを介して接続する場合、クライアントとデータベースサーバーの両方で SATMP を同じ値に設定する必要があります。

**注意**
SQL Anywhere バージョン 9 以前では、環境変数 ASTMP が SATMP に相当します。共有メモリを使用してバージョン 9 とバージョン 10 のソフトウェアに接続する場合は、SATMP と ASTMP 環境変数を設定して、テンポラリフォルダーとして (同じ) ロケーションを指定する必要があります。

データベースサーバーや UNIX 上のクライアントで作成されたテンポラリファイルのパーミッションを制限する場合は、この環境変数に次のリストにないディレクトリを設定します。

- */tmp*
- */tmp/.SQLAnywhere*
- TMP 環境変数の値 (設定している場合)
- TMPDIR 環境変数の値 (設定している場合)
- TEMP 環境変数の値 (設定している場合)
- 上記のいずれかのディレクトリを示すシンボリックリンク

上記のリストにないディレクトリが SATMP に設定されている場合、データベースサーバーは、現在のユーザーが所有し、パーミッションが 770、707、700 のいずれかに設定されているディレクトリで指定のディレクトリパスを検索します。パーミッションがそれ以外の値に設定されている場合は、パーミッションを 777 に設定してファイルを作成します。データベースサーバーは、検出された各ディレクトリについて、該当するパーミッション (それぞれ other、group、other +group に対応) をテンポラリファイルの作成に使用するパーミッションマスクから削除します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**警告**
上記のリストにないディレクトリを SATMP に設定すると、別の UNIX アカウントを使用するユーザーが共有メモリを介してデータベースサーバーに接続できなくなる場合があります。

UNIX での共有メモリ接続の保護については、SATMP 環境変数をユニークなディレクトリに設定 (UNIX)1381 ページを参照してください。

### 参照

- 「-dt dbeng12/dbsrv12 サーバーオプション」221 ページ
- 「TMP、TMPDIR、TEMP 環境変数」457 ページ
- 「Windows での環境変数の設定」443 ページ
- 「Unix および Mac OS X 環境変数」443 ページ
- 「異なるファイルの異なるデバイスへの配置」『SQL Anywhere サーバー SQL の使用法』
- 「テンポラリファイル用の十分なディスク領域があるかを確認する」1211 ページ
- 「sa_disk_free_space システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』

## SHLIB_PATH 環境変数 [HP-UX]

HP-UX 上で SQL Anywhere アプリケーションが必要とするライブラリの実行時の検索場所になるディレクトリを指定します。

### 構文

**SHLIB_PATH=***path-list*

### デフォルト

- */opt/sqlanywhere12/lib32* (32 ビットのプラットフォーム)
- */opt/sqlanywhere12/lib64* (64 ビットのプラットフォーム)

### 備考

インストーラーによって作成される *sa_config.sh* ファイルと *sa_config.csh* ファイルは、これを含むすべての環境変数を作成または変更するスクリプトです。

### 参照

- 「DYLD_LIBRARY_PATH 環境変数 [Mac OS X]」445 ページ
- 「LD_LIBRARY_PATH 環境変数 [Linux と Solaris]」445 ページ
- 「LIBPATH 環境変数 [IBM AIX]」446 ページ
- 「Unix および Mac OS X 環境変数」443 ページ

## SQLANY12 環境変数

SQL Anywhere 12 を含むディレクトリのロケーションを指定します。

### 構文

**SQLANY12=***directory-name*

### デフォルト

| オペレーティングシステム | ロケーション |
|---|---|
| Windows | *C:¥Program Files¥SQL Anywhere 12* |
| IBM AIX | */usr/lpp/sqlanywhere12* |
| Mac OS X | */Applications/SQLAnywhere12/System* |
| その他の UNIX オペレーティングシステム | */opt/sqlanywhere12* |

### 備考

この環境変数は必ず設定してください。これには、いくつかの理由があります。その 1 つとして、サンプルが SQL Anywhere アプリケーションを探すときに、この環境変数が必要となることがあります。

Windows では、SQLANY12 環境変数のロケーションがインストーラーによって設定されます。

UNIX では、これを含むすべての環境変数を作成または変更するスクリプトとして、*sa_config.sh* ファイルと *sa_config.csh* ファイルがインストーラーによって作成されます。

### 参照


- 「Windows での環境変数の設定」443 ページ
- 「Unix および Mac OS X 環境変数」443 ページ


## SQLANYSAMP12 環境変数

SQL Anywhere サンプルディレクトリのロケーションを指定します。

### 構文

**SQLANYSAMP12=***directory-name*

### デフォルト

| オペレーティングシステム | デフォルトのロケーション |
|---|---|
| Windows (Windows Mobile、Windows Vista、Windows 7、Windows Server 2008、Windows Server 2008 R2 を除く) | *C:¥Documents and Settings¥All Users ¥Documents¥SQL Anywhere 12¥Samples* |
| Windows Vista、Windows 7、Windows Server 2008、Windows Server 2008 R2 | *C:¥Users¥Public¥Documents¥SQL Anywhere 12¥Samples* |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オペレーティングシステム | デフォルトのロケーション |
|---|---|
| UNIX (単一ユーザーのインストール環境) | *$SQLANY12/samples* |

### 備考

Windows では、SQLANYSAMP12 環境変数のロケーションがインストーラーによって設定されます。Windows エクスプローラーウィンドウでこのディレクトリを開くには、**[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[サンプルアプリケーションとプロジェクト]** をクリックします。

Mac OS X を含む UNIX では、*sample_config* スクリプトは SQL Anywhere サンプルのユーザーごとのコピーを作成するために使用できます。このスクリプトを実行して、オペレーティングシステムに SQLANYSAMP12 環境変数のロケーションを設定する必要があります。

UNIX では、インストーラーは SQLANYSAMP12 環境変数を sa_config スクリプトと sample_config スクリプトに設定します。これは、マルチユーザーインストール環境の場合に便利です。シングルユーザーのインストール環境では、sa_config スクリプトは SQLANYSAMP12 を *$SQLANY12/samples* に設定します。

### 参照


- 「SQLANY12 環境変数」453 ページ
- 「Windows での環境変数の設定」443 ページ
- 「Unix および Mac OS X 環境変数」443 ページ


## SQLCONNECT 環境変数

データベースサーバーへの接続時に使用する追加の接続パラメーターを指定します。

### 構文

**SQLCONNECT=***parameter=value*; ...

### 備考

この文字列は、*parameter=value* 形式のパラメーター設定をセミコロンで区切ったリストで指定します。

接続文字列ですでに指定されている場合、SQLCONNECT 環境変数によって指定された接続パラメーターは使用されません。

サポートされている接続パラメーターの詳細については、「接続パラメーター」320 ページを参照してください。

> **警告**
> パスワードはプレーンテキストなので、それを SQLCONNECT 環境変数に含めることはセキュリティリスクとなります。

### 参照


- 「Windows での環境変数の設定」443 ページ
- 「Unix および Mac OS X 環境変数」443 ページ


## SQLPATH 環境変数

SQL スクリプトとヘルプファイルのロケーションを指定します。

### 構文

**SQLPATH=***path-list*

### 備考

Interactive SQL は、システムパスを検索する前に、SQLPATH に指定されているディレクトリでスクリプトファイルとヘルプファイルを検索します。

### 参照


- 「Windows での環境変数の設定」443 ページ
- 「Unix および Mac OS X 環境変数」443 ページ


## SQLREMOTE 環境変数

SQL Remote 中の FILE メッセージリンクのアドレスである、サブフォルダーを指定します。

### 構文

**SQLREMOTE=***path*

### 備考

SQL Remote 中の FILE メッセージリンクのアドレスは、SQLREMOTE 環境変数のサブフォルダーです。この環境変数には共有ディレクトリを指定してください。

Windows オペレーティングシステム (Windows Mobile を除く) では、SQLREMOTE 環境変数を設定する代わりに、*SQL Remote¥Directory* レジストリエントリを適切なルートディレクトリに設定することもできます。

### 参照


- 「Windows での環境変数の設定」443 ページ
- 「Unix および Mac OS X 環境変数」443 ページ


## SYBASE 環境変数

Adaptive Server Enterprise、Sybase Open Client、Sybase Open Server などの Sybase アプリケーションや DSEdit などのユーティリティのインストール用ホームディレクトリを指定します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 構文

**SYBASE=***directory-name*

### 備考

この環境変数が必要になるのは、その他の Sybase アプリケーションを使用する場合のみです。

### 参照


- 「Windows での環境変数の設定」443 ページ
- 「Unix および Mac OS X 環境変数」443 ページ


## TMP、TMPDIR、TEMP 環境変数

SQL Anywhere テンポラリファイルのロケーションを指定します。

### 構文

**TMP=***path*

**TMPDIR=***path*

**TEMP=***path*

### 備考

SQL Anywhere は、データベースサーバーが起動すると同時に作成され、データベースサーバーが停止すると消去されるテンポラリファイルのような、さまざまな操作のためのテンポラリファイルを作成する場合があります。その名前が示すように、テンポラリファイルとはデータベースサーバーの起動中に一時的に情報を保持するために使用されます。テンポラリファイルには、セッション間にわたって維持する必要がある情報は格納されません。

テンポラリファイルは、環境変数 TMP、TMPDIR、TEMP のいずれかで指定されたディレクトリに作成されます。これらの環境変数が重複指定されている場合は、TMP、TMPDIR、TEMP の中で最初に出現したものが採用されます。SATMP 環境変数を設定して、データベースサーバーとテンポラリディレクトリを必要とする SQL Anywhere コマンドラインユーティリティが使用するテンポラリファイルのロケーションを指定することもできます。

データベースサーバーが使用するテンポラリファイルのロケーションは、データベースサーバーの起動時に -dt サーバーオプションを使用して指定できます。テンポラリファイルのロケーションを指定せずにデータベースサーバーを起動した場合は、SQL Anywhere で環境変数が SATMP、TMP、TMPDIR、TEMP の順序のとおりにチェックされます。環境変数が定義されていない場合、テンポラリファイルは、Windows の場合は現在のディレクトリ、UNIX の場合は */tmp* ディレクトリに作成されます。

UNIX では、テンポラリディレクトリは共有メモリ接続を確立するために使用されます。Windows Mobile では、サーバーのテンポラリフォルダーとして使用するディレクトリをレジストリに指定できます。テンポラリディレクトリ値の設定については、「Windows Mobile でのレジストリ設定」466 ページを参照してください。

**注意**

**UNIX における旧バージョンのソフトウェアでの共有メモリ接続の使用**　SQL Anywhere バージョン 9 以前では、環境変数 ASTMP が SATMP に相当します。共有メモリを使用してバージョン 9 とバージョン 12 のソフトウェアに接続する場合は、SATMP と ASTMP 環境変数を設定して、テンポラリファイルのロケーションを指定する必要があります。

**参照**


- 「-dt dbeng12/dbsrv12 サーバーオプション」221 ページ
- 「SATMP 環境変数」451 ページ
- 「Windows での環境変数の設定」443 ページ
- 「Unix および Mac OS X 環境変数」443 ページ
- 「異なるファイルの異なるデバイスへの配置」『SQL Anywhere サーバー SQL の使用法』
- 「テンポラリファイル用の十分なディスク領域があるかを確認する」1211 ページ
- 「データベース ファイル」『SQL Anywhere 12 紹介』


# ファイルロケーションとインストール設定

SQL Anywhere をインストールすると、いくつかのディレクトリが作成されます。このディレクトリ内のファイルの中には、不可欠なものもあり、そうでないものもあります。この項ではディレクトリ構造について説明します。

1 つの製品として入手した場合も、他の製品の一部として入手した場合も、SQL Anywhere ソフトウェアは単一のインストールディレクトリにインストールされます。SQLANY12 環境変数は、インストールディレクトリのロケーションを指定します。「SQLANY12 環境変数」453 ページを参照してください。

## SQL Anywhere インストールディレクトリ

SQL Anywhere のインストールディレクトリには、次のようなものがあります。

- **『はじめにお読みください』**　*readme.txt* ファイルには、最新の情報が記載されています。

Windows Mobile 以外のプラットフォームの場合、インストールディレクトリの下に次のようなディレクトリがあります。

- **実行プログラムディレクトリ**　オペレーティングシステムプラットフォームごとに別々のディレクトリがあり、それぞれに設定ファイルとコンテキスト別のヘルプファイルが保存されています。

  Windows Mobile を除く Windows では、これらのファイルは *bin32* または *bin64* ディレクトリにインストールされます。UNIX を使用している場合、ファイルは *bin32* または *bin64* と *lib32* または *lib64* ディレクトリにインストールされます。

  使用しているオペレーティングシステムのバージョンに必要なディレクトリだけがあります。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

- ***java* ディレクトリ**　このディレクトリには JAR ファイルが保存されます。
- ***scripts* ディレクトリ**　scripts ディレクトリには、データベース管理ユーティリティによって使用され、サンプルとしても使用される SQL スクリプトがあります。
- ***¥SDK¥Include* ディレクトリ**　*¥SDK¥Include* ディレクトリには、SQL Anywhere 対応の C/C++ アプリケーションを開発するためのヘッダーファイルが保存されます。UNIX では、このディレクトリは *include* と呼ばれます。

### Windows Mobile ファイルのロケーション

Windows Mobile デバイスでは、すべてのファイルがインストールディレクトリ *¥Program Files ¥SQLAny12* に保存されます。ただし、DLL は *¥Windows* ディレクトリに保存されます。サブフォルダーは作成されません。

### UNIX ファイルのロケーション

言語リソースは *res* ディレクトリにインストールされ、共有オブジェクトは *lib32* または *lib64* ディレクトリにインストールされます。

### サンプルディレクトリ

SQL Anywhere をインストールするとき、サンプルのインストール先ディレクトリを選択できます。このマニュアルではそのロケーションを *samples-dir* と表します。%SQLANYSAMP12% 環境変数は、*samples-dir* のロケーションを指定します。「SQLANYSAMP12 環境変数」454 ページを参照してください。

Windows でサンプルにアクセスするには、**[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[サンプルアプリケーションとプロジェクト]** をクリックします。

次の表に、サポートされている各オペレーティングシステムのサンプルディレクトリのデフォルトロケーションと一般的なロケーションを示します。

| オペレーティングシステム | デフォルトのインストールロケーション | 一般的なインストール先 |
|---|---|---|
| Windows XP/2003 | *%ALLUSERSPROFILE%¥Documents ¥SQL Anywhere 12¥Samples* | C:¥Documents and Settings¥All Users¥Documents¥SQL Anywhere 12¥Samples [1] |
| Windows Vista、Windows 7、Windows Server 2008、Windows Server 2008 R2 | *%PUBLIC%¥Documents¥SQL Anywhere 12¥Samples* | C:¥Users¥Public¥Documents¥SQL Anywhere 12¥Samples |
| Windows Mobile | *¥Program Files¥SQLAny12* | |
| Mac OS X | */Applications/SQLAnywhere12/Samples* | |
| IBM AIX | */usr/lpp/sqlanywhere12/samples* | |

| オペレーティングシステム | デフォルトのインストールロケーション | 一般的なインストール先 |
|---|---|---|
| その他の UNIX オペレーティングシステム | */opt/sqlanywhere12/samples* | |

[1] SQL Anywhere サンプルディレクトリに Windows エクスプローラーでアクセスする場合、ロケーションは *C:¥Documents and Settings¥All Users¥「Shared Documents」¥SQL Anywhere 12/Samples* です。ただし、SQL Anywhere サンプルディレクトリにコマンドプロンプトでアクセスする場合のパスは、*C:¥Documents and Settings¥All Users¥「Documents」¥SQL Anywhere 12¥Samples* です。

# SQL Anywhere のファイル検索方法

クライアントライブラリとデータベースサーバーは、主に次の 2 つの理由からファイルを検索する必要があります。

- SQL Anywhere を実行するためには、DLL と初期化ファイルが必要です。不正な DLL が検索されると、バージョンミスマッチエラーが発生する可能性があります。
- INSTALL JAVA や LOAD TABLE など、SQL 文で指定して、ランタイムに検索する必要があるものがあります。

ファイル名を使用する SQL 文の例は次のとおりです。

- **INSTALL JAVA 文**　Java クラスを保存するファイル名です。
- **LOAD TABLE 文と UNLOAD TABLE 文**　データがロードまたはアンロードされるファイル名です。
- **CREATE DATABASE 文**　この文と、ファイルを作成できる同様の文には、ファイル名が必要です。

SQL Anywhere は簡易アルゴリズムを使用してファイルを検索する場合があります。それ以外の場合、より広範囲の検索が行われます。

## 簡単なファイル検索

多くの SQL 文 (LOAD TABLE や CREATE DATABASE など) では、ファイル名はデータベースサーバーの現在作業中のディレクトリに対する相対名として解釈されます。

また、データベースサーバーが起動され、データベースファイル名 (DatabaseFile (DBF) パラメーター) が提供されると、パスは現在作業中のディレクトリに対する相対名として解釈されます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## Windows での広範囲なファイル検索

Windows では、データベースサーバーと管理ユーティリティを含む SQL Anywhere プログラムは、DLL や共有ライブラリなど、必要なファイルをより広範囲に検索できます。この場合、SQL Anywhere プログラムは次の順番でファイルを検索します。

1. モジュールのディレクトリ (プログラムの実行ファイルまたはライブラリファイルがあるディレクトリ)。
2. 実行プログラムディレクトリ (プログラムの実行ファイルがあるディレクトリ)
3. SQL Anywhere インストールパスは単一のディレクトリであり、定義されている場合は %SQLANY12% 環境変数で指定されます。
4. フルパスが指定されます。パスが何も指定されない場合は、現在の作業ディレクトリが指定されます。
5. *%APPDATA%¥SQL Anywhere 12* ディレクトリ。
6. *%ALLUSERSPROFILE%¥SQL Anywhere 12* ディレクトリ。
7. ロケーションレジストリエントリ。
8. システムに応じたディレクトリ。このディレクトリには、一般的なオペレーティングシステムファイルが格納されているディレクトリが含まれます。たとえば Windows では、*Windows* ディレクトリや *Windows¥system32* ディレクトリになります。
9. PATH ディレクトリ。システムパスとユーザーパスのディレクトリを検索します。

**注意**
Windows では、前述のリストのディレクトリからの相対パスで次のパスが検索されます。

a. .
b. ..
c. *.¥bin32* と *..¥bin32* (32 ビットのプログラムのみ)
d. *.¥bin64* と *..¥bin64* (64 ビットのプログラムのみ)
e. *.¥java* (Java 関連のファイル)
f. *..¥java* (Java 関連のファイル)
g. *.¥scripts* (SQL スクリプトファイル)
h. *..¥scripts* (SQL スクリプトファイル)

## Windows Mobile での広範囲なファイル検索

Windows Mobile では、データベースサーバーと管理ユーティリティを含む SQL Anywhere プログラムは、DLL や共有ライブラリなど、必要なファイルをより広範囲に検索できます。この場合、SQL Anywhere プログラムは次の順番でファイルを検索します。

1. モジュールのディレクトリ (プログラムの実行ファイルまたはライブラリファイルがあるディレクトリ)。
2. 実行プログラムディレクトリ (プログラムの実行ファイルまたはライブラリがあるディレクトリ)。
3. フルパスが指定されます。
4. ロケーションレジストリエントリ。
5. システムに応じたディレクトリ。これには、*Windows* など、一般的なオペレーティングシステムファイルがあるディレクトリが含まれます。

**注意**
Windows Mobile では、前述のリストのディレクトリからの相対パスで次のパスが検索されます。

a. .
b. ..
c. *.¥bin32*
d. *..¥bin32*
e. *.¥java* (Java 関連のファイル)
f. *..¥java* (Java 関連のファイル)
g. *.¥scripts* (SQL スクリプトファイル)
h. *..¥scripts* (SQL スクリプトファイル)

## UNIX での広範囲なファイル検索

UNIX では、データベースサーバーと管理ユーティリティを含む SQL Anywhere プログラムは、DLL や共有ライブラリなど、必要なファイルをより広範囲に検索できます。この場合、SQL Anywhere プログラムは次の順番でファイルを検索します。

1. 実行プログラムのパス (判別できる場合)。
2. SQL Anywhere インストールパスは単一のディレクトリであり、定義されている場合は $SQLANY12 環境変数で指定されます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

3. パスなし (現在の作業ディレクトリ)。
4. *$HOME/.sqlanywhere12* ディレクトリ。
5. PATH 環境変数。
6. LIBPATH 環境変数
   - LD_LIBRARY_PATH (Linux と Solaris)
   - LD_LIBRARY_PATH と SHLIB_PATH (HP-UX)
   - LIBPATH (IBM AIX)
   - DYLD_LIBRARY_PATH (Mac OS X)

**注意**
UNIX では、前述のリストのディレクトリからの相対パスで次のパスが検索されます。

1. .
2. ..
3. *./bin32* と *../bin32* (32 ビットのプログラムのみ)
4. *./bin64* と *../bin64* (64 ビットのプログラムのみ)
5. *./lib32* と *../lib32* (32 ビットのプログラムのライブラリファイルのみ)
6. *./lib64* と *../lib64* (64 ビットのプログラムのライブラリファイルのみ)
7. *./java* (Java 関連のファイル)
8. *../java* (Java 関連のファイル)
9. *./scripts* (SQL スクリプトファイル)
10. *../scripts* (SQL スクリプトファイル)
11. *./res* (*.res* ファイル)
12. *../res* (*.res* ファイル)
13. *./tix* (*.tix* ファイル)
14. *../tix* (*.tix* ファイル)

# レジストリと INI ファイル

Windows オペレーティングシステム (Windows Mobile 以外) の場合、SQL Anywhere でレジストリ設定が使用されます。UNIX の場合、これらの設定は初期化ファイルに保存されています。

これらの設定はソフトウェアのインストール中に行われるので、通常はユーザーがレジストリや初期化ファイルにアクセスする必要はありません。ここでは、オペレーティング環境を変更するユーザーのために説明します。

SQL Anywhere によって使用される *.ini* ファイルの内容は、ファイル難読化ユーティリティを使用した単純暗号化によって難読化できます。 「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。

**警告**
SQL Anywhere データソースだけを使用している場合を除き、UNIX 上でファイル難読化ユーティリティ (dbfhide) を使用して、システム情報ファイル (デフォルトのファイル名は *.odbc.ini*) に単純暗号化を追加しないでください。他のデータソース (Mobile Link 同期など) を使用する予定の場合、システム情報ファイルの内容を難読化すると、他のドライバーが正しく機能しなくなることがあります。

## 現在のユーザーとローカルマシン設定

オペレーティングシステムによっては、2 つのレベルのシステム設定が存在することがあります。一部の設定は個々のユーザーに固有であり、そのユーザーがログオンしたときだけ使用できます。これらの設定を「現在のユーザー」設定と呼びます。また、コンピューター全体に関連し、すべてのユーザーが使用できるものを、「ローカルマシン」設定と呼びます。ローカルマシン設定を変更するには、コンピューターの管理者パーミッションを取得する必要があります。

SQL Anywhere は、現在のユーザー設定とローカルマシン設定の両方を許可します。たとえば、Windows XP では、これらの設定は *HKEY_CURRENT_USER* キーと *HKEY_LOCAL_MACHINE* キーにそれぞれ保管されています。

### 現在のユーザーを優先

現在のユーザーとローカルマシンレジストリの両方に設定がある場合は、現在のユーザー設定がローカルマシン設定より優先されます。

### ローカルマシン設定が必要なとき

SQL Anywhere プログラムを「サービス」として実行する場合は、設定がローカルマシンレベルで行われていることを確認してください。

サービスは、コンピューター全体を停止しないかぎり、コンピューターからログオフしても特別なアカウントで実行を継続できます。サービスは、個々のアカウントに依存しないようにできます。そのため、ローカルマシン設定にアクセスすることが必要です。

SQL Anywhere プログラムの他に、一部の Web サーバーがサービスとして実行されます。そのような Web サーバーで Apache や Microsoft IIS を使用するには、ローカルマシン設定をしてください。

一般的には、ローカルマシン設定をおすすめします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## *.ini* ファイルの内容の非表示

♦ **ファイルの内容を非表示にするには、次の手順に従います。**

多くの場合、SQL Anywhere では *.ini* ファイルに特定の名前が付けられていると想定します。名前が重要なファイル (*saldap.ini* など) に単純暗号化を追加する場合、元のファイルのコピーを別の名前を付けて保存してください。元のファイルのコピーを保存していない場合、ファイルがいったん難読化されると、その内容を変更できません。次の手順では、*.ini* ファイルに単純暗号化を追加する方法について説明します。

1. ファイルを別の名前で保存します。

   ```
   rename saldap.ini saldap.ini.org
   ```

2. ファイル難読化ユーティリティを使用してファイルを難読化し、難読化されたファイルに必要なファイル名を付けます。

   ```
   dbfhide saldap.ini.org saldap.ini
   ```

3. ファイルシステムまたはオペレーティングシステムの保護により *saldap.ini.org* ファイルを保護するか、安全な場所に保存します。

   *saldap.ini* ファイルに変更を加えるには、*saldap.ini.org* ファイルを編集し、手順 2 を繰り返します。

> **警告**
> SQL Anywhere データソースだけを使用している場合を除き、UNIX 上でファイル難読化ユーティリティ (dbfhide) を使用して、システム情報ファイル (デフォルトのファイル名は *.odbc.ini*) に単純暗号化を追加しないでください。他のデータソース (Mobile Link 同期など) を使用する予定の場合、システム情報ファイルの内容を難読化すると、他のドライバーが正しく機能しなくなることがあります。

**参照**

- 「ファイル非表示ユーティリティ (dbfhide)」953 ページ

## レジストリ構造

Windows (Windows Mobile 以外) では、レジストリエディターでレジストリに直接アクセスできます。SQL Anywhere レジストリエントリは、以下のロケーションにある *HKEY_CURRENT_USER* または *HKEY_LOCAL_MACHINE* キーに保管されています。

```
Software
  Sybase
    SQL Anywhere
      12.0
    Sybase Central
      6.1.0
```

> **警告**
> レジストリは、ユーザー自身の責任で変更してください。システムをバックアップしてからレジストリを変更することをおすすめします。

## インストール時のレジストリ設定

**警告**
特定の Intel ストレージドライバーを使用するシステムの停電に対する堅牢性を高めるには、特定のレジストリエントリを設定します。このパラメーターを設定しないと、停電発生時にデータが失われたり、データベースが破損したりする可能性があります。Intel ストレージドライバーの堅牢性を高めるには、次の手順に従います。『SQL Anywhere サーバー プログラミング』を参照してください。

Windows では、*HKEY_LOCAL_MACHINE¥Software¥Sybase* レジストリ内の次の設定は、インストールプログラムによって行われます。次のリストは、レジストリ設定の一部を示します。

- ***SQL Anywhere¥12.0¥Location*** このエントリには、SQL Anywhere ソフトウェアのインストールディレクトリが設定されます。次に例を示します。

  Location "C:¥Program Files¥SQL Anywhere 12"

- ***SQL Anywhere¥12.0¥Samples Location*** このエントリには、サンプルプログラムのインストールディレクトリが設定されます。次に例を示します。

  Samples Location "C:¥Documents and Settings¥All Users¥Documents¥SQL Anywhere 12¥Samples¥"

- ***SQL Anywhere¥12.0¥Online Resources*** このエントリにはオンラインリソースのロケーションが設定されます。次に例を示します。

  Online Resources "C:¥Program Files¥SQL Anywhere 12¥support¥ianywhere.html"

- ***SQL Anywhere¥12.0¥Language*** このエントリには、メッセージとエラーに使用される現在の言語を示す 2 文字のコードが設定されます。次に例を示します。

  Language "EN"

  言語は、インストール中に指定された:言語選択に基づいて設定されます。「ロケール言語」485 ページを参照してください。

- ***Sybase Central¥6.1.0¥Language*** このエントリには、メッセージとエラーに使用される現在の言語を示す 2 文字のコードが設定されます。次に例を示します。

  Language "EN"

  このエントリは Sybase Central によって使用されます。言語は、インストール中に指定された:言語選択に基づいて設定されます。「ロケール言語」485 ページを参照してください。

## Windows Mobile でのレジストリ設定

Windows Mobile でサーバーのテンポラリフォルダーとして使用するディレクトリは、次のレジストリ値で指定できます。

*HKEY_CURRENT_USER¥Software¥Sybase¥SQL Anywhere¥12.0¥TempFolder*

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

*TempFolder* は、使用するテンポラリディレクトリのパスに設定されます。サーバーは次のいずれかを実行します。

●指定のディレクトリが存在する場合、そのディレクトリを使用する。

●指定のディレクトリがなく、親ディレクトリが存在する場合、指定のディレクトリを作成する。

指定のディレクトリがなく、作成もできない場合、データベースサーバーでは次のように処理されます。

●*¥Temp* ディレクトリが存在する場合、そのディレクトリを使用する。

●*¥Temp* ディレクトリが存在しない場合、そのディレクトリを作成する。

*¥Temp* ディレクトリが存在せず、作成もできない場合、サーバーは現在のディレクトリを使用します。

# 国際言語と文字セット

国際化とは、ソフトウェアの指定言語やソフトウェアを実行しているオペレーティングシステムに関係なく、ソフトウェアが各種言語とそれに適した文字セットに対処できるようにすることです。SQL Anywhere は完全に国際化に対応しています。次に、要求されたり使用されたりすることが多い機能について説明します。

● **Unicode のサポート**　SQL Anywhere の Unicode サポートは次のとおりです。

○クライアントでは、ODBC、OLE DB、ADO.NET、JDBC の SQL Anywhere クライアントライブラリの UTF-16 をサポートしています。

○UTF-8 の Unicode 文字データの格納に NCHAR データ型を使用できます。

○CHAR データ型では UTF-8 エンコードを使用できます。

● **コードページと文字セット**　SQL Anywhere データベースサーバーと関連ツールは、Windows (ANSI/ISO)、UTF-8、UNIX のコードページと文字セットをサポートしています。

● **照合**　SQL Anywhere では、照合アルゴリズムとして、SQL Anywhere 照合アルゴリズム (SACA) と、International Components for Unicode (ICU) を使用した Unicode 照合アルゴリズム (UCA) の 2 つをサポートしています。

ICU の詳細については、「ICU とは何か、いつ必要になるか」470 ページを参照してください。

SACA を使用すると、ソートが高速、簡潔、実用的になりますが、言語的な正確さが低下します。UCA を使用すると、言語的な処理は正確になりますが、記憶領域の要件と実行時間が多少増加します。「照合」487 ページを参照してください。

高度なソートおよび比較機能として、SQL Anywhere には SORTKEY および COMPARE 関数が用意されています。これらの関数は、辞書や電話帳並みに言語的に高度なソート機能を実現します。必要に応じて、大文字小文字とアクセント記号を区別しないソートや比較ができます。 「SORTKEY 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』と「COMPARE 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

SQL Anywhere には、文字カラムで SORTKEY ベースのソートを自動的に使用する設計機能も用意されています。sort_collation データベースオプションを使用すると、文字カラムに ORDER BY が指定されたときに使用されるソート順序を指定できます。文字カラムのソートキーの格納に計算カラムも使用できるので、ORDER BY が指定されるたびに計算カラムを計算する必要はありません。「sort_collation オプション」678 ページを参照してください。

- **文字セット変換**　SQL Anywhere では、サーバーシステムとクライアントシステムとの間でデータの文字セットエンコードが変換され、複数の異なる文字セットを使用している環境でもデータの整合性が維持されます。「文字セット変換」478 ページを参照してください。

- **識別子**　SQL Anywhere では、ほとんどのシングルバイト文字やマルチバイト文字を含む識別子を引用符で囲まなくても使用できます。例外は、スペースや句読表記記号です。

- **通貨**　通貨記号は、ユーロ記号も含めて、ソートの対象になります。SQL Anywhere では、通貨の書式サポートはありません。

- **日付と時刻のフォーマット**　SQL Anywhere では太陽暦を採用し、日付と時刻の設定用にさまざまなフォーマットを用意しています。カスタムフォーマットは、date_format、time_format、timestamp_format の各データベースオプションを使用して実現できます。date_format と timestamp_format オプションのデフォルトは、ISO 互換の日付形式 YYYY-MM-DD です。SQL Anywhere には CONVERT 関数が用意されており、日付と時刻の出力フォーマットを一般的な各種フォーマットに変換できます。次の項を参照してください。

  - 「date_format オプション」613 ページ
  - 「time_format オプション」696 ページ
  - 「timestamp_format オプション」698 ページ
  - 「CONVERT 関数 [データ型変換]」『SQL Anywhere サーバー SQL リファレンス』

**参照**

- 「名前を付けた照合データベースの作成」503 ページ
- 「推奨文字セットと照合」506 ページ

# SQL Anywhere のローカライズ版

ローカライズとは、製品を目的のロケールの言語や文化に適用させることです。ロケールとは、通常は言語と国／地域の組み合わせです。ローカライズは、包装、インストーラー、マニュアル、ソフトウェアのユーザーインターフェイス、エラー／警告／情報メッセージなど、多くのコンポーネントに反映されます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

SQL Anywhere ソフトウェアは、次の 5 言語にローカライズされています。

- 英語
- フランス語
- ドイツ語
- 日本語
- 中国語 (簡体文字)

使用言語はインストール時に選択します。

マニュアルは、英語版、ドイツ語版、日本語版、簡体字中国語版が用意されています。

Windows の場合、**[スタート]** メニューを使用して、インストールされた言語と英語を切り替えることができます。言語選択ユーティリティ (dblang) を使用すると、追加で導入した言語を含む使用可能なすべての言語に設定し直すことができます。次の項を参照してください。


- 「Windows での配備ソフトウェアのローカライズ」470 ページ
- 「言語選択ユーティリティ (dblang)」983 ページ
- 「管理ツールの言語設定の変更」904 ページ


次の表では、各言語のサポート状況をオペレーティングシステムプラットフォーム別に示します。

| プラットフォーム | 英語 | フランス語 | ドイツ語 | 日本語 | 中国語 (簡体文字) |
|---|---|---|---|---|---|
| Windows | ○ | ○ | ○ | ○ | ○ |
| Windows Mobile | ○ | ○ | ○ | ○ | ○ |
| Linux | ○ | | ○ | ○ | ○ |
| UNIX | ○ | | | | |
| Mac OS X | ○ | | | | |

## ソフトウェアとマニュアルの完全ローカライズ

SQL Anywhere の Windows 版は次の言語で提供されており、開発、配備、管理に適しています。

- 英語
- フランス語
- ドイツ語
- 日本語
- 中国語 (簡体文字)

英語版、ドイツ語版、日本語版、簡体字中国語版では、次に示す SQL Anywhere のすべてのコンポーネントがローカライズされています。

●インストーラー

●マニュアルとコンテキスト別のヘルプ

●ソフトウェア

フランス語版では、インストーラー、ソフトウェア、コンテキスト別のヘルプがローカライズされています。

次のコンポーネントはローカライズされません。英語版のみで使用できます。

- InfoMaker
- Sybase PowerDesigner Physical Data Model

## Windows での配備ソフトウェアのローカライズ

上記 5 言語の他に、SQL Anywhere には次の言語の配備ソフトウェアリソースが用意されています。

- イタリア語
- 韓国語
- リトアニア語
- ポーランド語
- ブラジルポルトガル語
- ロシア語
- スペイン語
- 中国語 (繁体文字)
- ウクライナ語

配備ローカライズは、エンドユーザーに配備されることが多い一部のソフトウェアコンポーネントが対象となります。包装、マニュアル、管理ソフトウェア、開発ソフトウェア、およびインストールソフトウェアはローカライズされません。ローカライズされるソフトウェアコンポーネントは次のとおりです。

- データベースサーバーとクライアントライブラリ
- Mobile Link サーバーおよびクライアント
- SQL Remote クライアント
- コマンドラインツール (dbinit、dbunload など)

## ICU とは何か、いつ必要になるか

ICU (International Components for Unicode) は、IBM が開発および保守しているオープンソースライブラリです。ICU は、Unicode サポートを提供することによって、ソフトウェアの国際化を容易にします。SQL Anywhere は、ICU を使用して、特定の文字セット変換と照合オペレーションを実装しています。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### データベースサーバーで ICU が必要になるとき (Windows Mobile を除くすべてのプラットフォーム)

データベースサーバーで常に ICU を利用できることが理想的です。次の表に、ICU が必要になるときとその理由を示します。

| ICU が必要になるとき | 説明 |
| --- | --- |
| NCHAR 文字セットまたは CHAR 文字セットの照合として Unicode Collation Algorithm (UCA) が使用されている | UCA は ICU を必要とします。 |
| データベースの文字セットは UTF-8 でなく、マルチバイト文字セットである | データベースの文字セットから UTF-8 にパスワード変換するために必要です (データベースパスワードは内部的に UTF-8 で格納されています)。 |
| クライアントとデータベースの文字セットが異なり、いずれかがマルチバイト (UTF-8 を含む) である。これには、Unicode ODBC、OLE DB、ADO.NET、SQL Anywhere JDBC アプリケーションが含まれます。ICU のインストールされていないクライアントが使用するデータベース文字セットには関係ありません。 | マルチバイト文字セットの適切な変換を行うためには ICU が必要です。 |
| データベースの文字セットが UTF-8 でなく、CHAR と NCHAR 間の変換が必要である | データベースサーバーは、UTF-8 を別の文字セットに変換するために ICU を必要とします。 |
| Embedded SQL クライアントが UTF-8 以外の NCHAR 文字セットを使用する | データベースサーバーは、UTF-8 を別の文字セットに変換するために ICU を必要とします。Embedded SQL クライアントのデフォルトの NCHAR 文字セットは、最初のクライアントの CHAR 文字セットと同じです。これは、db_change_nchar_charset 関数を使用して変更できます。「db_change_nchar_charset 関数」『SQL Anywhere サーバー プログラミング』を参照してください。 |

| ICU が必要になるとき | 説明 |
| --- | --- |
| CSCONVERT または SORTKEY 関数が使用されている。CSCONVERT 関数は、上記の 3 番目の項目要件に準ずる文字セットを変換するために呼び出されます。 | マルチバイト文字セットの文字セット変換を行うためには ICU が必要です。SORTKEY のラベル用に SORTKEY を生成するには UCA が必要であるため、結果として ICU が必要になります。「CSCONVERT 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』と「SORTKEY 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |

### データベースサーバーで ICU が必要になるとき (Windows Mobile)

次の表に、Windows Mobile で ICU が必要になるときとその理由を示します。

| ICU が必要になるとき | 説明 |
| --- | --- |
| NCHAR 照合または CHAR 照合として UCA が使用されている | UCA は ICU を必要とします。 |
| SORTKEY 関数が使用されている | SORTKEY のラベル用に SORTKEY を生成するには UCA が必要であるため、結果として ICU が必要になります。「SORTKEY 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| CHAR 文字セットが OS 文字セットに一致しない | 文字セットが一致する場合であっても、NCHAR を使用しているとき、または CHAR 文字セットがマルチバイトであるときは、文字セット変換の処理を向上させるため、ICU の使用をおすすめします。 |

**注意**
ICU ライブラリをインストールしていない場合は、Windows Mobile の文字セットと一致する文字セットを使う照合を選択するか、データベースを作成するときに CHAR 照合として UTF8BIN 照合を選択することが必要です。また、データベースを作成するときに NCHAR 照合として UTF8BIN を選択することも必要になります。

### データベースサーバー上で ICU を使用しないで正しい文字セット変換を実行できるとき

データベースの文字セットとクライアントの文字セットの両方がシングルバイトであり、*sqlany.cvf* を利用できるとき (すべてのプラットフォーム)、またはオペレーティングシステムが変換をサポートするとき (Windows のみ) は、ICU を使用しないで正しい文字セット変換を行うことができます。これは、*sqlany.cvf* を利用できる場合、またはホストのオペレーティングシス

テムに適切な変換機能がインストールされている場合にかぎり、ICU を使用しなくてもシングルバイト間の変換が処理可能なためです。

### クライアントで ICU が必要になるとき (Windows Mobile を除くすべてのプラットフォーム)

Unicode のクライアントアプリケーションでは、使用するデータベースの文字セットにかかわらず、すべてのクライアントに ICU がインストールされている場合に、クライアントとデータベースサーバー間のパフォーマンスが向上する可能性があります。これは、必要な変換処理の一部がデータベースサーバーからクライアントに渡されて、負荷が軽減するからです。

また、Windows プラットフォームで ODBC を使用している場合は、ANSI アプリケーションであっても、クライアントに ICU をインストールしておくことが必要です。これは、ドライバーマネージャーが ANSI ODBC 呼び出しを Unicode ODBC 呼び出しに変換するからです。

## 文字セット

各ソフトウェアは、「文字セット」を使用します。文字セットは記号、文字、数字、スペースなどから成ります。"ISO-8859-1" は文字セットの例です。Latin1 とも呼ばれます。

文字を内部的に適切に表すため、各ソフトウェアは「エンコード」 (「文字コード」とも呼ばれる) を使用します。エンコードとは、各文字を 1 バイトまたは複数バイトの情報にマッピングする方法で、16 進数で表します。UTF-8 はエンコードの例です。

「文字セット」と「エンコード」は密接に関連しており、どちらも「エンコード」の意味で使用されることがあります。

「コードページ」は、エンコードの一形態です。コードページとは文字と数値表現とのマッピングのことで、通常、数値表現は 0 ～ 255 の整数です。コードページの例には、Windows コードページ 1252 があります。

このマニュアルでは、「エンコード」、「文字コード」、「文字セットエンコード」、「コードページ」を同じ意味で使用します。

データベースサーバーは、文字をソート (たとえば、名前をアルファベット順にリスト) するときに「照合」を使用します。照合は文字コード (文字と表現間のマッピング) と文字の「ソート順」の組み合わせです。各文字列にソート順が複数ある場合があります。たとえば、大文字／小文字を区別するソート順と大文字／小文字を区別しないソート順があります。また、言語間で同じ文字に対するソート順が異なる場合もあります。

文字は「フォント」を使って画面上に表示されます。これは文字セットの文字とその外観との間のマッピングです。フォントはオペレーティングシステムによって処理されます。

オペレーティングシステムは、「キーボードマッピング」を使って、キーボードのキーまたはキーの組み合わせを文字セットの文字にマッピングします。

## クライアント／サーバーコンピューティングにおける言語の問題

クライアントアプリケーションで作業するデータベースユーザーは、次のソースから文字列を参照したり、文字列にアクセスしたりする場合があります。

- **データベース内のデータ**　データベースには文字列やその他のテキストデータが格納されています。データベースサーバーは要求に応答するときに、これらの文字列を処理します。たとえば、データベースサーバーが、テーブルの N より後の文字で始まるすべての名前を表示するよう求められることがあります。この要求では、文字列比較を実行する必要がありますが、特定の文字セットのソート順序が想定されています。

- **データベースサーバーソフトウェアメッセージ**　アプリケーションによってデータベースエラーが引き起こされることがあります。たとえば、存在しないカラムを参照するクエリをアプリケーションが送信した場合です。この場合、データベースサーバーは警告かエラーメッセージを返します。このメッセージは「言語リソースライブラリ」に保持されます。これは SQL Anywhere が使用する DLL または共有ライブラリです。

- **クライアントアプリケーション**　クライアントアプリケーションのインターフェイスはテキストを表示します。また、内部でテキストを処理できます。

- **クライアントソフトウェアメッセージ**　クライアントライブラリは、データベースサーバーと同じ言語を使用してクライアントアプリケーションにメッセージを提供します。

- **オペレーティングシステム**　クライアントとサーバーのオペレーティングシステムは、メッセージを提供したりテキストを処理したりします。

環境を適切に動作させるためには、テキストの入力箇所のすべてで統合的に機能しなければなりません。大まかに言うと、すべてユーザーの言語および文字セット、またはそのいずれかで動作させてください。

## シングルバイト文字セット

多くの言語では文字の数はシングルバイトの文字セットで扱える程度です。このような文字セットでは、各文字はシングルバイト (2 桁の 16 進数) で表されます。

シングルバイトのセットでは最大 256 文字を表すことができます。アクセント記号の付いた文字を含め、国際的に使用するすべての文字を保持できるシングルバイト文字セットはありません。この問題は、1 つ以上の国の言語に適する文字セットを記述するコードページのセットの開発により解決されました。たとえば、コードページ 1253 にはギリシャ語の文字セットが含まれており、コードページ 1252 には西ヨーロッパ言語の文字セットが含まれています。コードページは多数あり、その名前も多数あります。上記の例は、Windows のコードページです。

### 上方ページと下方ページ

わずかな例外はありますが、文字 0 ～ 127 はすべてのコードページで共通です。この範囲の文字のマッピングを 「ASCII」 文字セットと呼びます。これには、英語のアルファベットの大文字と小文字、共通の句読表記記号、数字が含まれます。この範囲は「7 ビット範囲」 (127 までの文字を表すのに 7 ビットしか必要ないため) または「下方ページ」と呼ばれます。128 ～ 255 ま

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

での文字は「拡張文字」、または「上方コードページ文字」と呼ばれ、コードページ間で異なります。

英語のアルファベット文字だけを使用する場合は、各コードページの ASCII 部分 (0 ～ 127) のみで表せるため、コードページの互換性の問題が起こることはほとんどありません。しかし、その他の文字を使用すると、非英語環境ではよく起こることですが、データベースとアプリケーションが異なるコードページを使用している場合に、問題が起こる可能性があります。

たとえば、UTF-8 文字セットを使用するデータベースが cp1252 データを含むファイルからテーブルをロードするときに、LOAD TABLE 文でエンコードが cp1252 に指定されていなかったとします。エンコードが指定されていないので、データのエンコードは UTF-8 であると想定され、文字変換は実行されないので、cp1252 エンコードがデータベースに直接格納されます。つまり、cp1252 では 16 進数の 80 で表されているユーロ記号が UTF-8 に変換されません。UTF-8 のユーロ記号は E2 82 AC の 3 バイトシーケンスで表されますが、この例の場合はデータベースに 80 として格納されます。後でアプリケーションからデータを要求されたとき、データベースサーバーはデータを UTF-8 からクライアントの文字セットに変換しようとします。この変換により、文字の破損が発生します。

## マルチバイト文字セット

言語によっては (日本語や中国語など)、256 文字よりもはるかに多い文字があります。この場合はシングルバイトを使用しては表示できないので、マルチバイトのエンコードを使用する必要があります。さらに、多くの言語の文字を単一の文字セットで表現するために、マルチバイトの文字セットよりも多くの文字を使う文字セットも存在します。この例として、UTF-8 が挙げられます。

マルチバイト文字セットは「可変幅」で、いくつかの文字はシングルバイト、他はダブルバイトなどになります。

### 参照

- 「SQL Anywhere 照合アルゴリズム (SACA)」488 ページ

### 例

たとえば、コードページ 932 (日本語) の文字の長さは 1 バイトまたは 2 バイトです。最初のバイト (「リードバイト」とも呼ばれる) の値が 16 進数値 ¥x81 ～ ¥x9F または ¥xE0 ～ ¥xFC (10 進数値 129 ～ 159 または 224 ～ 252) の範囲にある場合、その文字は 2 バイト文字であり、直後のバイト (「フォローバイト」とも呼ばれる) と併せて文字が成立します。フォローバイトとは、最初のバイト以外のすべてのバイトのことです。

最初のバイトがリードバイトの範囲外にある場合、その文字はシングルバイト文字であり、次のバイトは次の文字の最初のバイトになります。

## Windows 環境の ANSI コードページと OEM コードページ

Windows の場合、2 つのコードページが使用されています。Windows グラフィカルユーザーインターフェイスを使用するアプリケーションでは、Windows コードページが使用されます。Windows コードページには、ISO 文字セットおよび ANSI 文字セットとの互換性があります。このようなコードページは、しばしば 「ANSI コードページ」 と呼ばれます。

Windows で動作する文字モードアプリケーション (コマンドプロンプトウィンドウを使用するアプリケーション) は DOS で使用されていたコードページを使用します。これは歴史的な理由から 「OEM コードページ」 (Original Equipment Manufacturer) と呼ばれます。

SQL Anywhere は、OEM と ANSI コードページの両方に基づいた照合をサポートします。OEM 照合は互換性確保のためにサポートされていますが、新しいデータベースでは使用しないでください。

**参照**

- 「サポートされている照合と代替照合」492 ページ

## SQL Anywhere データベース内での文字セット

SQL Anywhere データベースでは、文字データの格納に 1 つまたは 2 つの文字セット (エンコード) を使用できます。CHAR データ型 (CHAR、VARCHAR、LONG VARCHAR など) では、シングルバイト文字セットまたはマルチバイト文字セットが使用されます。UTF-8 も使用できます。NCHAR データ型 (NCHAR、NVARCHAR、LONG NVARCHAR など) では、UTF-8 が使用されます。

LOAD TABLE などの SQL 文や CSCONVERT、TO_CHAR、TO_NCHAR などの関数は、パラメーターとして 「db_charset」 と 「nchar_charset」 を使用して、それぞれデータベースの CHAR 文字セットと NCHAR 文字セットを参照します。

**参照**

- 「文字データ型」『SQL Anywhere サーバー SQL リファレンス』
- 「LOAD TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「CSCONVERT 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』
- 「TO_CHAR 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』
- 「TO_NCHAR 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』

## サポートされている文字セット

SQL Anywhere では、数多くの文字セットとラベルをサポートしており、その対象は増え続けています。文字セットエンコードラベルは、さまざまな名前やラベルが知られています。SQL Anywhere でサポートされている文字セットのリストを表示するには、次のコマンドを実行します。

```
dbinit -le
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

出力の各行には、指定された文字セットエンコードで最も一般的なラベルがカンマで区切られた形式で表示されます。各行の最初のラベルは、その文字セットエンコードで優先される SQL Anywhere 名です。その他のラベルは、別の機関、組織、または標準によって使用されるもので、IANA (Internet Assigned Numbers Authority)、MIME (Multipurpose Internet Mail Extensions)、ICU (International Components for Unicode)、Java、および ASE (Adaptive Server Enterprise) があります。

目的の文字セットが見つからない場合は、次のコマンドを実行して、あまり一般的でない文字セットを含むリストを参照できます。

```
dbinit -le+
```

文字セットエンコードのラベルを指定すると、SQL Anywhere によってラベルの検索が行われます。場合によっては、他の機関が異なる文字セットに対して同じラベルを使用することがあります。SQL Anywhere は、可能なかぎり、あいまいさを解決しようとします。たとえば、あいまいなラベルで設定されている文字を参照する JDBC アプリケーションは、Java 標準のラベルに解決されます。あいまいさを回避するため、常に SQL Anywhere ラベルを使用することをおすすめします。文字セットエンコードのラベルの詳細については、次の URL を参照してください。

dbinit -le オプションによって返される文字セットエンコードのラベルのほかに、次の文字セットエイリアスも使用できます。

- **os_charset**　データベースサーバーをホストしているオペレーティングシステムが使用する文字セットのエイリアス。
- **char_charset**　データベースが使用する CHAR 文字セットのエイリアス。
- **nchar_charset**　データベースが使用する NCHAR 文字セットのエイリアス。

特定の文字セットまたはラベルがサポートされているかどうかを簡単に確認するには、CSCONVERT 関数を使用します。

**参照**

- 「CSCONVERT 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』

## 文字セットに関する質問とその回答

次の表に、各質問とその回答が記述されている参照先を示します。

| 質問 | 参照先 |
| --- | --- |
| データベースで使用する照合を決定する方法は？ | 「照合」 |
| ソフトウェア、特に SQL Anywhere における文字の表示方法は？ | 「文字セット」 |
| SQL Anywhere が提供する照合の種類は？ | 「照合の考慮事項」 |

| 質問 | 参照先 |
| --- | --- |
| SQL Anywhere がサポートする文字セットエンコードは？ | 「サポートされている文字セット」 |
| クライアントコンピューターとデータベースとで使用文字セットが違う。クライアントとサーバー間で文字を正しくやり取りする方法は？ | 「文字セット変換」 |
| 接続文字列で使用できる文字セットは？ | 「接続文字列と文字セット」 |
| 既存のデータベースの照合順を変更する方法は？ | 「データベースの照合を変更する」 |

# 文字セット変換

SQL Anywhere は、文字セットまたはコードページの異なる場所で同じ文字を表している文字セット間で、文字セット変換を実行できます。これを可能にするには、文字セット間にある程度の互換性が必要になります。たとえば、EUC-JIS と cp932 との間では文字セット変換を実行できますが、EUC-JIS と cp1252 との間では実行できません。

SQL Anywhere では、IBM が開発および保守している International Components for Unicode (ICU) オープンソースライブラリを使用して文字セット変換を実装しています。

**参照**

- 「データ型間の比較」『SQL Anywhere サーバー SQL リファレンス』

# 接続文字列と文字セット

使用している文字セットがクライアント間で異なる場合、接続文字列の文字セット変換はたいへん困難になります。接続文字列は、データベースサーバーを検出または起動するために、クライアントライブラリによって解析されますが、解析はデータベースサーバーの文字セットや言語が未知のまま実行されます。

インターフェイスライブラリは、接続文字列を次のように解析します。

1. 接続文字列は、*keyword=value* の組み合わせに分解されます。これは、CommLinks (LINKS) パラメーターの前後に中カッコ {} を使用しないかぎり、文字セットにかかわらず実行されます。中カッコの代わりにカッコ () を使用することをおすすめします。中カッコは、一部のマルチバイト文字セットで有効な「フォローバイト」 (最初のバイト以外のバイト) です。

2. データベースサーバーが検出されます。サーバー名に対して文字セット変換は実行されません。クライアントとデータベースサーバーとで文字セットが異なる場合、サーバー名に拡張文字が使用されているとサーバーが見つからないことがあります。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

クライアントとサーバーを異なるオペレーティングシステムやロケールで実行している場合は、サーバー名に 7 ビットの ASCII 文字を使用してください。

3. DatabaseName (DBN) 接続パラメーターまたは DatabaseFile (DBF) 接続パラメーターは、クライアントの文字セットからデータベースサーバーの文字セットに変換されます。

4. データベースが検出されると、残りの接続パラメーターがデータベースの文字セットに変換されます。

## SQL 文と文字セット

SQL Anywhere サーバーの文字セット変換によって、すべての SQL 文は解析と実行の前にデータベース文字セットに変換されます。この変換による副作用として、SQL 文に含まれるデータベース文字セットに変換できない文字は置換文字に変換されます。任意の Unicode 文字が含まれた SQL 文は、次のいずれかの方法で実行できます。

●UNISTR 関数を使用して Unicode 文字値を指定する

●ホスト変数を使用して Unicode 文字値を指定する

●データベースの文字セットとして UTF-8 を使用する

UTF8BIN を CHAR 照合として選択した場合、データベースの文字セットは UTF-8 です。CHAR 照合として UCA を選択した場合、UTF-8 をエンコードとして選択できます。

Unicode 照合アルゴリズム (UCA) では、高度な比較、ソート、大文字と小文字の変換ができますが、パフォーマンスが低下する場合があります。UTF8BIN は必要とする領域が少なく、高速ですが、ソート順と比較はバイナリで実行されます。SQL 文で Unicode 文字を使用する場合は、CHAR 照合に UTF8BIN を指定しますが、ソートと比較では UCA の一部の機能しか必要ありません。必要な場合のみ、SORTKEY と COMPARE 関数を使用して UCA を使用します。

**参照**


- 「SORTKEY 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』
- 「COMPARE 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』
- 「Unicode 照合アルゴリズム (UCA)」489 ページ
- 「SQL Anywhere 照合アルゴリズム (SACA)」488 ページ
- 「接続文字列として渡される接続パラメーター」31 ページ


## データを表示した場合の予期しない記号のトラブルシューティング

Interactive SQL などのクライアントアプリケーションを使用してデータを表示すると、四角形、矢印、疑問符などの予期しない記号が文字としてデータに表示される場合があります。

予期しない記号が表示される最も一般的な理由は、データの表示に使用されたフォントが該当の文字をサポートしていないことです。このような問題は、Unicode のフォントに変更することで解決できます。クライアントアプリケーションのフォントを変更できない場合は、代わりにオペレーティングシステムのデフォルトのフォントを変更できます。

また、データベースに格納されている基本となるデータに問題がある場合にも、予期しない記号がクライアントアプリケーションで表示される可能性があります。たとえば、データをデータベースに追加する場合に文字セット変換が必要で、元の文字セットの一部の文字と対応する文字がデータベースの文字セットになかった場合、その文字の代わりに置換文字が挿入されます。

たとえば、Windows では、標準の英語フォント Tahoma は日本語の文字をサポートしていません。データベースの文字セットが cp932 で、データベースに日本語のデータが含まれている場合、データベースにクエリを実行すると、結果セットの日本語の文字は小さい四角記号で表示されます。Interactive SQL では、**[ツール]** » **[オプション]** » **[SQL Anywhere]**/**[Ultra Light]** » **[結果]** » **[フォント]** をクリックして、Arial Unicode MS や Lucida Sans Unicode などのユニコードフォントを指定すると、**[結果]** ウィンドウ枠のフォントを変更できます。ユニコードフォントは多言語の文字を表示できるため、このような場合の使用をおすすめします。

クライアントアプリケーションがフォント設定の変更を許可しない場合は、オペレーティングシステムのデフォルトのフォントを使用している可能性があります。このような場合、デフォルトのシステムフォントをユニコードフォントに変更する方法については、オペレーティングシステムのマニュアルを参考にしてください。

**参照**

- 「損失を伴う変換および置換文字」『SQL Anywhere サーバー SQL リファレンス』
- 「Interactive SQL」 840 ページ

## 外国語での大文字と小文字の区別

SQL Anywhere では常に、システムビュー名やカラム名などの識別子の「大文字と小文字を維持」し、「大文字と小文字を区別しません」。名前は作成時の大文字と小文字のまま格納されますが、識別子へのアクセスは大文字と小文字の区別なしで行われます。

たとえば、システムビューの名前は大文字 (SYSDOMAIN、SYSTAB など) で格納されますが、アクセスは大文字と小文字の区別なく行うことができます。したがって、次の 2 つの文は同等です。

```
SELECT * FROM systab;
SELECT * FROM SYSTAB;
```

照合では、大文字と小文字が同等と定義されています。ただし、一部の照合では、識別子の大文字と小文字の区別を前提とする場合、特別な注意が必要です。たとえば、トルコ語の照合では、予期できない複雑なエラーが発生するような大文字と小文字の変換動作があります。最も一般的なエラーは、**I** または **i** という文字を含むシステムオブジェクトが見つからないというものです。「トルコ語文字セットと照合」 509 ページを参照してください。

## 文字セット変換とクライアント API

マルチ文字セット環境で作業する場合、文字セット変換の問題が発生すると、その問題がどこで発生したかの特定が困難である可能性があります。クライアント API で文字セット変換の問題が発生した場合は、文字セット変換を制御するデータベースと接続のオプションやプロパティを確認してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

変換の問題が発生する可能性がある箇所は、2 つのカテゴリに分けられます。1 つ目のカテゴリは、誤ったフォーマットのデータをクライアント API に送信したことに関係します。このカテゴリの問題は、Unicode API では発生しませんが、他のすべてのクライアント API で発生する可能性があり、データは無効になります。

問題の 2 つ目のカテゴリは、最終的な文字セットまたはいずれかの中間文字セットに対応するものがない文字に関係します。この場合、置換文字が使用されます。これは損失を伴う変換と呼ばれ、どのクライアント API でも発生する可能性があります。データベース文字セットに UTF-8 を使用するようにデータベースを設定すると、損失を伴う変換を回避できます。「損失を伴う変換および置換文字」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

### 文字セット変換に影響するオプションとプロパティの設定

接続に影響するデータベースオプションとデータベースや接続のプロパティは、通常、完全な接続文字列で使用できます。ただし、PROPERTY、DB_PROPERTY、CONNECTION_PROPERTY などのシステム関数を使用して設定を問い合わせることもできます。次に例を示します。

| クエリ | 説明 |
|---|---|
| SELECT PROPERTY( 'CharSet' ); | データベースサーバーのオペレーティングシステムの文字セットを返します。 |
| SELECT DB_PROPERTY( 'CharSet' ); | データベースの CHAR 文字セットを返します。 |
| SELECT DB_PROPERTY( 'NcharCharSet' ); | データベースの NCHAR 文字セットを返します。 |
| SELECT DB_PROPERTY( 'MultibyteCharSet' ); | CHAR データでマルチバイト文字セットを使用するかどうか (On=使用、Off=使用しない) を返します。 |
| SELECT CONNECTION_PROPERTY( 'CharSet' ); | クライアントの CHAR 文字セットを返します。 |
| SELECT CONNECTION_PROPERTY( 'NcharCharSet' ); | クライアントの NCHAR 文字セットを返します。 |
| SELECT CONNECTION_PROPERTY( 'on_charset_conversion_failure' ); | on_charset_conversion_warning オプションの値を返します。 |

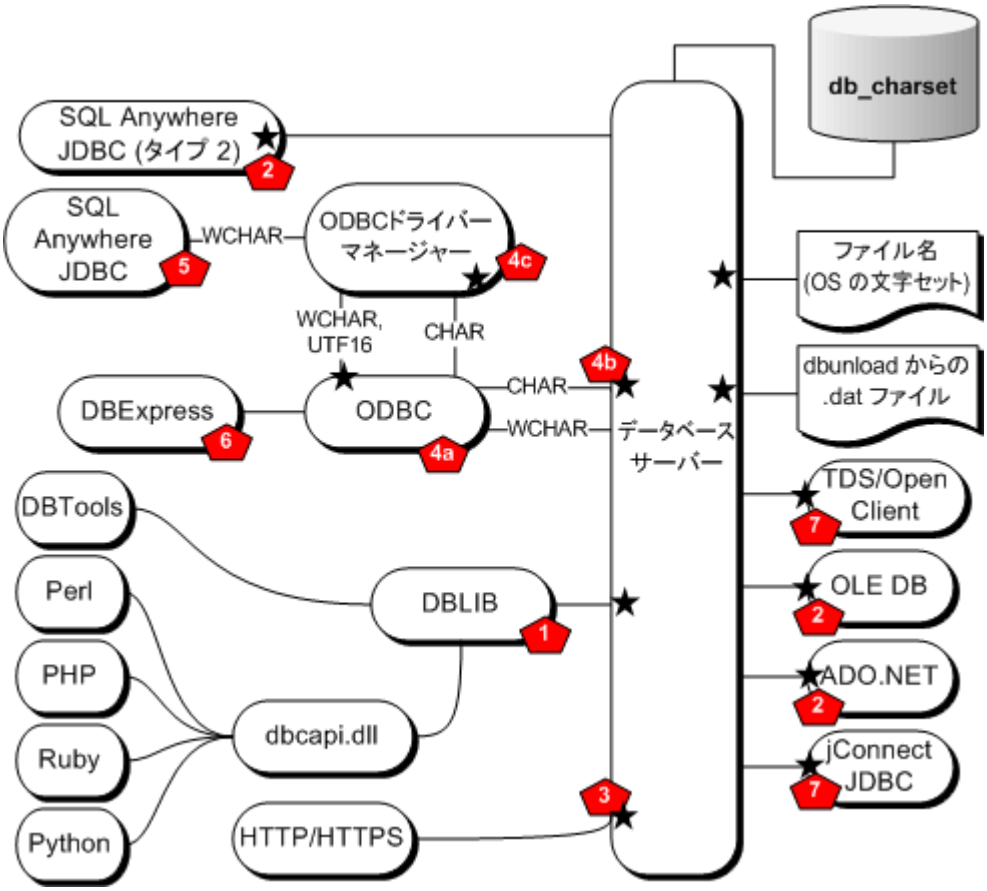
### 参照


- 「PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』
- 「DB_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』
- 「CONNECTION_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』
- 「on_charset_conversion_failure オプション」654 ページ

### クライアント API と文字セット変換のポイント

次の図は、クライアント API がデータベースサーバーと対話するときに文字セット変換が行われる可能性がある場所を示しています。

図の黒い星印は、文字セット変換が行われる可能性がある場所を示しています。図の数字は、図の後にある詳細な説明の番号に対応しています。




- **1 - DBLIB**　デフォルトで DBLIB CHAR 文字セットと NCHAR 文字セットがクライアントのオペレーティングシステムの文字セットになります。CHAR 文字セットは、BINARY と NCHAR を除くすべての文字列データに使用されます。NCHAR 文字セットは、NCHAR 型のホスト変数とホストパラメーター (これらは DBTools または DBCAPI インターフェイスでは使用できません) に使用されます。データベースサーバーでは、すべての文字セット変換が実行されます。

  CHAR 文字セットと NCHAR 文字セットは、それぞれ CharSet 接続パラメーターと db_change_nchar_charset 関数を使用して設定できます。「CharSet (CS) 接続パラメーター」325 ページと「db_change_nchar_charset 関数」『SQL Anywhere サーバー プログラミング』を参照してください。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

- **2 - Unicode API**　Unicode API を使用すると、ドライバーは文字データに Unicode UTF-16 エンコードを使用し、Unicode データとデータベース文字セット間の変換を行います。ドライバーは Unicode ホストパラメーターを NCHAR 文字セット (UTF-8) に変換してからデータベースサーバーに送信します。NCHAR として記述された結果セットのカラムは、NCHAR 文字セット (UTF-8) でフェッチされ、受信後に Unicode に変換されます。CharSet 接続パラメーターを設定すると損失を伴う変換が発生する可能性があるため、このパラメーターの設定はおすすめしません。「損失を伴う変換および置換文字」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

- **3 - HTTP/HTTPS**　「HTTP サービス (SQL Anywhere をサーバーとして使用) : 」HTTP サービスの場合、2 つのタイプの要求 (URL コード化された要求と multipart/form-data 要求) があります。

  **URL コード化要求**　要求は application/x-www-form-urlencoded の形式であり、変数はキーと値のペアで渡されます。データベースサーバーは % でコード化されたデータ (UTF-8 またはデータベース文字セット) を復号化し、キーと値のペアをデータベース文字セットに変換します。処理された値は HTTP_VARIABLE 関数を使用して抽出できます。この関数で @BINARY または @TRANSPORT 属性を使用すると、% で復号化され、文字セットが変換されていない値または未加工の HTTP (transport) 値をそれぞれ返すことができます。

  **Multipart/form-data 要求**　要求はバイナリと見なされます。つまり、@BINARY または @TRANSPORT 属性を使用すると、同じ値が返されます。

  HTTP サービス要求の文字セット変換は、sa_set_http_option システムプロシージャーを使用して設定できる HTTP オプション CharsetConversion と AcceptCharset によって制御されます。「sa_set_http_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

  要求や応答の文字セット変換に影響する可能性があるその他の設定には、優先文字セットの指定に使用される Accept-Charset ヘッダーとコンテンツの文字セットの指定に使用される Content-Type ヘッダーがあります。

  「HTTP ストアドプロシージャー (SQL Anywhere をクライアントとして使用) : 」HTTP ストアドプロシージャーでは、CHAR データはデータベース文字セットで送信されます。パラメーターが NCHAR 型の場合、すべてのデータは UTF-8 として送信されます (すべての CHAR パラメーターは UTF-8 に変換されます)。要求によって、優先文字セットとしてデータベース文字セットを指定する Accept-Charset HTTP ヘッダーが送信されます。優先文字セットのリストには、常に UTF-8 が含まれます。応答の Content-Type ヘッダーにデータベース文字セットではない文字セットが指定されている場合、応答はクライアントによってデータベース文字セットに変換されます。

- **4a - Unicode エントリポイントを使用する ODBC**　ODBC アプリケーションは Unicode エントリポイントを使用する場合、Unicode クライアント API と見なされます。WCHAR データは Unicode API の場合と同じ方法で処理されます。ODBC アプリケーションが Unicode エントリポイントと CHAR (ANSI) データを使用する場合、データはデータベース文字セットであると見なされます。1 つのアプリケーションで CHAR データと WCHAR データを混在させることはおすすめしません。

- **4b - ANSI エントリポイントを使用する ODBC**　ODBC アプリケーションは ANSI エントリポイントを使用する場合、ANSI クライアント API と見なされます。デフォルトで CHAR 文字

セットがクライアントオペレーティングシステムの文字セットになります。CHAR 文字セットは CharSet 接続パラメーターを使用して変更できます。「CharSet (CS) 接続パラメーター」325 ページを参照してください。

ANSI データの変換は、データベースサーバーによって実行されます。WCHAR ホスト変数にフェッチすると、データベースサーバーで CHAR データの損失を伴う変換が発生する可能性があります。1 つのアプリケーションで CHAR データと WCHAR データを混在させることはおすすめしません。

- **4c - ODBC ドライバーマネージャー**　一部の ODBC ドライバーマネージャーは、すべての CHAR データを WCHAR データに変換してから Unicode エントリ ポイントを呼び出します。

- **5 - SQL Anywhere JDBC ドライバー**　SQL Anywhere JDBC ドライバーは、文字セット変換を実行せず、ODBC ドライバーの Unicode エントリポイントと WCHAR 型を使用します。これは、Unicode エントリポイントを使用する ODBC で説明した内容と同じです。

- **6 - Borland DBExpress**　Borland DBExpress ドライバーは、ODBC を ANSI クライアント API として使用します。デフォルトで CHAR 文字セットがクライアントオペレーティングシステムの文字セットになります。CharSet 接続パラメーターを使用してこれを変更できますが、データが正しくなくなる可能性があるためおすすめしません。

- **7 - TDS クライアント**　Sybase Open Client や jConnect などの TDS クライアントは、接続時にデータベースサーバーとネゴシエートし、クライアントで文字セット変換を実行することを選択します。ネゴシエーション中、データベースサーバーは文字セット変換を実行しないよう指示されます。

**参照**

- 「データ型変換」『SQL Anywhere サーバー SQL リファレンス』
- 「損失を伴う変換および置換文字」『SQL Anywhere サーバー SQL リファレンス』
- 「NCHAR から CHAR への変換」『SQL Anywhere サーバー SQL リファレンス』
- 「SQL Anywhere C API のサポート」『SQL Anywhere サーバー プログラミング』
- 「Perl DBI サポート」『SQL Anywhere サーバー プログラミング』
- 「Python サポート」『SQL Anywhere サーバー プログラミング』
- 「SQL Anywhere PHP 拡張」『SQL Anywhere サーバー プログラミング』
- 「SQL Anywhere の Ruby API サポート」『SQL Anywhere サーバー プログラミング』
- 「Sybase Open Client のサポート」『SQL Anywhere サーバー プログラミング』
- 「HTTP Web サーバーとしての SQL Anywhere」『SQL Anywhere サーバー プログラミング』
- 「OLE DB」『SQL Anywhere サーバー プログラミング』
- 「SQL Anywhere .NET データプロバイダー」『SQL Anywhere サーバー プログラミング』
- 「ODBC サポート」『SQL Anywhere サーバー プログラミング』
- 「JDBC サポート」『SQL Anywhere サーバー プログラミング』

# ロケール

データベースサーバーとクライアントライブラリはどちらも、「ロケール定義」を使用して、言語と文字セット環境を認識します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

アプリケーションのロケールまたはクライアントのロケールは、データベースサーバーへの要求時にクライアントまたはクライアントライブラリによって使用され、返す結果に使用される文字セットと、エラーメッセージ、警告、その他のメッセージの言語を決定します。データベースサーバーは自身のロケールをアプリケーションのロケールと比較し、必要な文字セット変換を判断します。サーバー上のデータベースによって、ロケール定義が異なる場合があります。また、クライアントによってロケールが異なる場合もあります。

ロケールは次のコンポーネントで構成されています。

- **言語**　ISO 639-1 標準の値を使用した 2 文字の文字列です (たとえばドイツ語は DE)。データベースサーバーとクライアントのどちらにも、自身のロケールに対する言語の値があります。

  データベースサーバーは、ロードする言語ライブラリをロケール言語で判断します。データベースの作成時に照合が指定されていない場合、データベースサーバーは使用する照合の判断に文字セットと併せてロケール言語も使用します。

  クライアントライブラリでは、ロードする言語ライブラリをロケール言語で判断し、データベースからの要求もロケール言語で判断します。「ロケール言語」485 ページを参照してください。

- **文字セット**　文字セットとは、使用しているコードページまたはエンコードのことです。クライアントとサーバーのどちらにも文字セットの値がありますが、両者が異なる場合もあります。異なる場合は、文字セット変換を実行して相互運用性を確保します。「ロケール文字セット」487 ページを参照してください。

## ロケール言語

ロケール言語は、クライアントアプリケーションのユーザーによって使用される言語、またはデータベースサーバーのユーザーによって使用されることが予測される言語です。ロケール設定の確認方法については、「ロケール情報」501 ページを参照してください。

クライアントライブラリとデータベースサーバーはどちらも同じ方法でロケールの言語コンポーネントを特定します。

1. SALANG 環境変数が指定されている場合は、その値を使用します。「SALANG 環境変数」450 ページを参照してください。

2. Windows では、SALANG 環境変数がない場合、SQL Anywhere 言語レジストリのエントリを確認します。「インストール時のレジストリ設定」466 ページを参照してください。

3. オペレーティングシステムの言語設定を確認します。

4. 上記の設定で言語を特定できなかった場合は、デフォルトで英語になります。

### 言語ラベルの値

次の表は、有効な言語ラベルの値と対応する ISO 639 言語コードを示します。

| 言語 | ISO 639-1 言語コード | 言語ラベル | 代替ラベル |
|---|---|---|---|
| アラビア語 | AR | arabic | なし |
| チェコ語 | CS | czech | なし |
| デンマーク語 | DA | danish | なし |
| オランダ語 | NL | dutch | なし |
| 英語 | EN | us_english | english |
| フィンランド語 | FI | finnish | N/A |
| フランス語 | FR | french | N/A |
| ドイツ語 | DE | german | なし |
| ギリシャ語 | EL | greek | なし |
| ヘブライ語 | HE | hebrew | なし |
| ハンガリー語 | HU | hungarian | N/A |
| イタリア語 | IT | italian | N/A |
| 日本語 | JA | japanese | N/A |
| 韓国語 | KO | korean | N/A |
| リトアニア語 | LT | lithuanian | なし |
| ノルウェー語 | NO | norwegian | norweg |
| ポーランド語 | PL | polish | なし |
| ポルトガル語 | PT | portuguese | portugue |
| ロシア語 | RU | russian | N/A |
| 中国語 (簡体文字) | ZH | chinese | simpchin |
| スペイン語 | ES | spanish | なし |
| スウェーデン語 | SV | swedish | なし |
| タイ語 | TH | thai | N/A |
| 中国語 (繁体文字) | TW | tchinese | tradchin |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 言語 | ISO 639-1 言語コード | 言語ラベル | 代替ラベル |
|---|---|---|---|
| トルコ語 | TR | turkish | N/A |
| ウクライナ語 | UK | ukrainian | なし |

## ロケール文字セット

アプリケーションとサーバーのロケール定義のいずれにも文字セットがあります。アプリケーションは、データベースサーバーから文字列を要求するときに文字セットを使用します。データベースサーバーは、データベースの文字セットをアプリケーションの文字セットと比較し、文字セット変換が必要かどうかを判断します。データベースサーバーがデータベースとクライアント間の文字セット変換を実行できない場合は、接続が失敗します。

1. SACHARSET 環境変数が設定されている場合、その値で文字セットが特定されます。「SACHARSET 環境変数」449 ページを参照してください。

   データベースサーバーが SACHARSET を使用するのは、照合の指定がない状態で新しいデータベースを作成する場合だけです。

2. 接続文字列で文字セットが指定されている場合は、その文字セットが使用されます。「CharSet (CS) 接続パラメーター」325 ページを参照してください。

3. Sybase Open Client アプリケーションは、*Sybase* リリースディレクトリの *locales* サブフォルダーの *locales.dat* ファイルを確認します。

4. オペレーティングシステムの文字セット情報によって、次の方法でロケールを特定します。

   ●Windows オペレーティングシステムの場合、最新の Windows ANSI コードページが使用されます。

   ●UNIX プラットフォームの場合、LC_ALL、LC_MESSAGES、LC_CTYPE、LANG の各ロケール環境変数がこの順序で確認されます。この中で最初に設定されていた環境変数の値が、文字セットの決定に使用されます。文字セットをオペレーティングシステムから特定できなかった場合は、デフォルトで iso_1 (Windows コードページ 28591、ISO 8859-1 Latin I、ISO 8859-1 Latin-1、iso_8859-1:1987 とも呼ばれる) が使用されます。

5. それ以外のプラットフォームでは、デフォルトでコードページ cp1252 が使用されます。

ロケール設定の確認方法については、「ロケール情報」501 ページを参照してください。

## 照合

照合とは、特定の文字セットまたはエンコードに対する文字のソートおよび比較の方法です。SQL Anywhere では、照合アルゴリズムとして、SQL Anywhere 照合アルゴリズム (SACA) と Unicode 照合アルゴリズム (UCA) の 2 つをサポートしています。SACA を使用すると、ソートが

高速、簡潔、実用的になりますが、言語的な正確さが低下します。UCA を使用すると、言語的な処理は正確になりますが、記憶領域の要件と実行時間が多少増加します。

この項では、提供される照合と、照合の使い分けについて説明します。

特定の照合を使用してデータベースを作成する方法については、「名前を付けた照合データベースの作成」503 ページと「初期化ユーティリティ (dbinit)」958 ページを参照してください。

照合の適合化構文を使用した UCA 照合のカスタマイズについては、「照合の適合化オプション」496 ページを参照してください。

## SQL Anywhere 照合アルゴリズム (SACA)

SQL Anywhere 照合アルゴリズム (SACA) は、シングルバイト文字とマルチバイト文字の実用的な比較、ソート、大文字小文字変換を提供します。このアルゴリズムは、必要とする領域が少なく、高速です。インデックスなどの文字列は、マッピング後も元の文字列と長さが同じです。比較、ソート、大文字小文字変換のマッピングでは、文字列の各バイト値ごとの簡単なテーブルルックアップを使用しています。

### シングルバイト文字セット

典型的なシングルバイト文字セットの照合では、各文字のアクセント記号付きとアクセント記号なしのあらゆる形が同じ値にマッピングされており、照合ではアクセント記号は区別されません。同じ文字のアクセント記号付き形とアクセント記号なし形は同じとみなされ、ソートされると隣り合わせになります。

この照合では、アクセント記号を維持したまま、大文字と小文字が変換されます。

### マルチバイト文字セット

マルチバイト文字セットの場合、リードバイトが 256 個の異なる値にマッピングされています。フォローバイトはバイナリ値として比較されます。

マルチバイト文字セットのほとんどの照合では、文字セットエンコードにより、文字はリードバイトによって識別される 256 バイト単位のページにグループ化されるので、このマッピング方法で実用的なソート結果が得られます。これらのページと各ページに含まれる文字は、該当する文字セットとして合理的な順序で配置されています。通常、この照合では文字セットにおけるページの順序 (リードバイト) が保持されます。ページによっては、他の特徴に基づいてソートされます。たとえば、日本語コードページ 932 用の 932JPN 照合では、全角文字 (漢字) と半角文字 (カタカナ) をグループ分けしています。

大文字小文字変換は、7 ビットの英文字についてのみ提供されています。

### UTF-8 文字セット

UTF-8 はマルチバイト文字セットです。各文字は 1 ～ 4 バイトで構成されます。SQL Anywhere では、UTF-8 文字のソート用に UTF8BIN 照合を提供しています。

UTF8BIN では、リードバイトが 256 の異なる値にマッピングされており、フォローバイトはバイナリ値として比較されます。UTF-8 の文字表現方法と 256 というマッピングの数的制限に起

因して、同じ文字のアクセント記号付き形とアクセント記号なし形など、関連文字をグループ化できません。ソートの基準は基本的にバイナリです。

大文字小文字変換は、7 ビットの英文字についてのみサポートされています。

## Unicode 照合アルゴリズム (UCA)

Unicode 照合アルゴリズム (UCA) は、Unicode 文字セット全体のソートに使用するアルゴリズムです。これにより、言語的に正しい比較、ソート、大文字小文字変換が実現されます。UCA は Unicode 規格の一部として開発されました。SQL Anywhere では、IBM が開発および保守している International Components for Unicode (ICU) オープンソースライブラリを使用して UCA を実装しています。

**注意**
デフォルトの UCA ソート順により、ほとんどの言語のほとんどの文字が適切な順序でソートされます。ただし、同じ文字を使用する言語間でソートや比較にさまざまな違いがあるため、UCA ですべての言語について適切なソート順が得られるわけではありません。そのため、ICU は UCA を調整できる構文を提供しています。「照合の適合化オプション」496 ページを参照してください。

UCA を使用すると、少ない領域と時間で高度な比較、ソート、大文字小文字変換を実現できます。

マッピング後の文字列は元の文字列より長くなります。このアルゴリズムは、複雑な文字を的確に処理できます。

SQL Anywhere 照合アルゴリズム (SACA) とは異なり、Unicode 照合アルゴリズム (UCA) はシングルバイトの UTF-8 文字セットにのみ使用され、各文字を 1 つまたは複数の属性で区別します。文字の場合、属性は基底文字、アクセント記号、大文字小文字です。

文字以外の場合、通常は基底文字だけが属性になります。

UCA は、次の方法で文字を比較します。

- 基底文字を比較します。文字列の基底文字が他の文字列と異なる場合は、その時点で比較が完了します。アクセント記号や大文字小文字の違いは考慮されません。

- データベースでアクセント記号の違いが区別される場合は、アクセント記号が比較されます。アクセント記号が異なる場合、その時点で比較が完了します。大文字小文字の違いは考慮されません。

- データベースで大文字と小文字が区別される場合は、各文字の大文字と小文字が比較されます。

元の文字列の値が同じとみなされるのは、基底文字、アクセント記号、大文字小文字がまったく同じ場合だけです。

### 例

UCA を使用して次の表の第 1 カラムの文字列を比較するとします。後続のカラムには各文字列の 3 つの属性が記載されています。基底文字は同じで、違うのはアクセント記号と大文字小文字だけです。

| 文字列 | 基底文字 | アクセント記号 | 大文字小文字 |
|---|---|---|---|
| noel | noel | なし、なし、なし、なし | 小、小、小、小 |
| noël | noel | なし、なし、アクセント記号、なし、 | 小、小、小、小 |
| Noel | noel | なし、なし、なし、なし | 大、小、小、小 |
| Noël | noel | なし、なし、アクセント記号、なし、 | 大、小、小、小 |

次の表は、UCA を使用した場合にアクセント記号と大文字小文字の区別により可能な 4 つの条件によるソート順を示します。

| アクセント記号の区別 | 大文字と小文字の区別 | ORDER BY による結果 | 説明 |
|---|---|---|---|
| N | N | Noel、noël、Noël、noel (順不同) | ●アクセント記号は区別しない<br>●大文字と小文字も区別しない<br>●すべての値が同じとみなされる<br>●順序は要素数 4 の集合内でランダム |
| Y | N | Noel、noel (順不同)、その後に noël、Noël (順不同) | ●アクセント記号なしが先、アクセント記号ありが後。したがって ë の前に e がくる<br>●大文字と小文字は区別しない。N と n は 2 つの間でランダム |
| N | Y | Noel、Noël (順不同)、その後に noël、noel (順不同) | ●大文字が先、小文字が後。したがって n の前に N がくる<br>●アクセント記号は区別しない。e と ë は 2 つの間でランダム |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| アクセント記号の区別 | 大文字と小文字の区別 | ORDER BY による結果 | 説明 |
|---|---|---|---|
| Y | Y | Noel<br>noel<br>Noël<br>noël | ●アクセント記号なしが先、アクセント記号ありが後。したがって ë の前に e がくる<br>●大文字が先、小文字が後。したがって n の前に N がくる |

## SQL Anywhere データベースでの照合

### CHAR 照合

CHAR データ型 (CHAR、VARCHAR、LONG VARCHAR など) の照合では、SQL Anywhere 照合アルゴリズム (SACA) が使用される場合と Unicode 照合アルゴリズム (UCA) が使用される場合があります。どちらの場合も、使用される照合は CHAR 照合と呼ばれます。

### NCHAR 照合

NCHAR データ型 (NCHAR、NVARCHAR、LONG NVARCHAR など) の照合では、Unicode 照合アルゴリズム (UCA) が使用される場合と UTF8BIN 照合が使用される場合があります。UTF8BIN 照合では SQL Anywhere 照合アルゴリズム (SACA) が使用されます。

### 大文字小文字とアクセント記号の区別

作成時に大文字小文字を区別するよう指定されなかった SQL Anywhere データベースでは、大文字と小文字は区別されません。区別されるようにするには、該当するオプションを指定します。データベースの作成後、データベースを再構築しないで大文字と小文字が区別されるようにすることはできません。

データベースが大文字と小文字を区別するかどうかによって、SACA 照合でも UCA 照合でも大文字と小文字を区別するかどうかが決まり、それにより CHAR 照合と NCHAR 照合で大文字と小文字を区別するかどうかが決まります。

作成時にアクセント記号を区別するよう指定されなかった SQL Anywhere データベースでは、アクセント記号は区別されません。区別されるようにするには、該当するオプションを指定します。データベースの作成後、データベースを再構築しないでアクセント記号が区別されるようにすることはできません。

データベースがアクセント記号を区別するかどうかは、UCA 照合にのみ影響を与えます。UCA 照合が CHAR 照合や NCHAR 照合で使用されているかどうかには関係ありません。CHAR 照合と NCHAR 照合のどちらにも SACA 照合を使用することを選択した場合、アクセント記号に関するオプションに効力はありません。アクセント記号の区別は SACA 照合の属性であり、該当するオプションをデータベース作成時に使用して指定することはできません。

## サポートされている照合と代替照合

次の表は、SQL Anywhere バージョン 8 で採用され、SORTKEY 関数と COMPARE 関数で使用できる互換性の照合を示します。

SQL Anywhere バージョン 10 では、SORTKEY 関数と COMPARE 関数の実装が ICU (International Components for Unicode) と UCA (Unicode 照合アルゴリズム) に変更され、照合名は UCA 照合に対応しています。SQL Anywhere バージョン 10 以降でのソートと比較は、その前のバージョンの SQL Anywhere と異なる場合があります。

| 照合ラベル | 説明 |
|---|---|
| 874THAIBIN | Code Page 874、Windows Thai、ISO8859-11、バイナリ順序 |
| 932JPN | Code Page 932、日本語シフト JIS、Microsoft 拡張文字付き |
| 936ZHO | Code Page 936、中国語 (簡体文字)、PRC GBK 2312-80 8 ビットコード |
| 949KOR | Code Page 949、韓国語 KS C 5601-1987 コード、完成型 |
| 950ZHO_HK | Code Page 950、中国語 (繁体文字)、Big 5 コード (HKSCS を含む) |
| 950ZHO_TW | Code Page 950、中国語 (繁体文字)、Big 5 コード |
| 1250LATIN2 | Code Page 1250、Windows Latin 2、中央／東ヨーロッパ言語 |
| 1250POL | Code Page 1250、Windows Latin 2、ポーランド語 |
| 1251CYR | Code Page 1251、キリル語 |
| 1252LATIN1 | Code Page 1252、Windows Latin 1、西ヨーロッパ言語 |
| 1252NOR | Code Page 1252、Windows Latin 1、ノルウェー語 |
| 1252SPA | Code Page 1252、Windows Latin 1、スペイン語 |
| 1252SWEFIN | Code Page 1252、Windows Latin 1、スウェーデン語／フィンランド語 |
| 1253ELL | Code Page 1253、Windows ギリシア語、ISO8859-7 拡張付き |
| 1254TRK | Code Page 1254、Windows Latin 5、トルコ語、ISO 8859-9 拡張付き |
| 1254TRKALT | Code Page 1254、Windows トルコ語、拡張付き ISO8859-9、I-dot と I-no-dot は同じ |
| 1255HEB | Code Page 1255、Windows ヘブライ語、ISO8859-8 拡張付き |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 照合ラベル | 説明 |
|---|---|
| 1256ARA | Code Page 1256、Windows アラビア語、ISO8859-6 拡張付き |
| 1257LIT | Code Page 1257、リトアニア語 |
| EUC_CHINA | 中国語 (簡体文字) の GB 2312-80 コード |
| EUC_JAPAN | 日本語の EUC JIS X 0208-1990 と JIS X 0212-1990 コード |
| EUC_KOREA | 韓国語の KS C 5601-1992 コード、Johab (組合型) |
| EUC_TAIWAN | 台湾語の Big 5 コード |
| ISO1LATIN1 | ISO8859-1、ISO Latin 1、西ヨーロッパ言語、Latin 1 順序 |
| ISO9LATIN1 | ISO8859-15、ISO Latin 9、西ヨーロッパ言語、Latin 1 順序 |
| ISO_1 | ISO8859-1、Latin 1、西ヨーロッパ言語 |
| ISO_BINENG | バイナリ順序、英語 ISO/ASCII 7 ビット文字ケースマッピング |
| UCA | 標準のデフォルト UCA 照合 |
| UTF8BIN | UTF-8、Unicode 用 8 ビットマルチバイトエンコード、バイナリ順序 |

### 代替照合

代替照合は、古いバージョンの SQL Anywhere との互換性や、その他の特殊な用途で使用できます。サポートされている代替照合の一覧を表示するには、次のコマンドを実行します。

```
dbinit -l+
```

### Adaptive Server Enterprise の照合

次の表は、SORTKEY 関数などの機能で使用するためにサポートされている Adaptive Server Enterprise 照合を示します。

| 説明 | 照合名 | 照合 ID |
|---|---|---|
| デフォルト Unicode マルチ言語 | default | 0 |
| CP 850 代替言語：アクセント記号なし | altnoacc | 39 |
| CP 850 代替言語：小文字優先 | altdict | 45 |

| 説明 | 照合名 | 照合 ID |
|---|---|---|
| CP 850 西ヨーロッパ言語：大文字／小文字の区別、優先設定なし | altnocsp | 46 |
| CP 850 スカンジナビア語辞書順 | scandict | 47 |
| CP 850 スカンジナビア語：大文字／小文字の区別、優先設定なし | scannocp | 48 |
| GB ピンイン | gbpinyin | なし |
| バイナリソート | binary | 50 |
| Latin-1 英語、フランス語、ドイツ語辞書順 | dict | 51 |
| Latin-1 英語、フランス語、ドイツ語、大文字／小文字の区別なし | nocase | 52 |
| Latin-1 英語、フランス語、ドイツ語、大文字／小文字の区別、優先設定なし | nocasep | 53 |
| Latin-1 英語、フランス語、ドイツ語、アクセント記号なし | noaccent | 54 |
| Latin-1 スペイン語辞書順 | espdict | 55 |
| Latin-1 スペイン語、大文字／小文字の区別なし | espnocs | 56 |
| Latin-1 スペイン語、アクセント記号なし | espnoac | 57 |
| ISO 8859-5 ロシア語辞書順 | rusdict | 58 |
| ISO 8859-5 ロシア語、大文字／小文字の区別なし | rusnocs | 59 |
| ISO 8859-5 キリル語辞書順 | cyrdict | 63 |
| ISO 8859-5 キリル語、大文字／小文字の区別なし | cyrnocs | 64 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 説明 | 照合名 | 照合 ID |
|---|---|---|
| ISO 8859-7 ギリシャ語辞書順 | elldict | 65 |
| ISO 8859-2 ハンガリー語辞書順 | hundict | 69 |
| ISO 8859-2 ハンガリー語、アクセント記号なし | hunnoac | 70 |
| ISO 8859-2 ハンガリー語、大文字／小文字の区別なし | hunnocs | 71 |
| ISO 8859-5 トルコ語辞書順 | turdict | 72 |
| ISO 8859-5 トルコ語、アクセント記号なし | turnoac | 73 |
| ISO 8859-5 トルコ語、大文字／小文字の区別なし | turnocs | 74 |
| CP 874 (TIS 620) タイ語辞書順 | thaidict | 1 |
| ISO 14651 順序付け標準 | 14651 | 22 |
| シフト JIS バイナリ順 | sjisbin | 179 |
| Unicode UTF-8 バイナリソート | utf8bin | 24 |
| EUC JIS バイナリ順 | eucjisbn | 192 |
| GB2312 バイナリ順 | gb2312bn | 137 |
| CP932 MS バイナリ順 | cp932bin | 129 |
| Big5 バイナリ順 | big5bin | 194 |
| EUC KSC バイナリ順 | eucksc bn | 161 |

**参照**

- 「COMPARE 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』
- 「SORTKEY 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』
- 「ICU とは何か、いつ必要になるか」470 ページ

# 照合の考慮事項

データベースを作成するとき、SQL Anywhere ではオペレーティングシステムの言語と文字セットの設定に基づいてデフォルト照合を選択できます。通常、デフォルト照合は適切な選択ですが、用意されている多数の照合の中からニーズに合った照合を明示的に選択することもできます。SQL Anywhere が特定の言語に対して複数の照合をサポートする場合もあります。

データベースのデータに適した文字セットとソート順を使用する照合を選択してください。文字列のソートや比較を詳細に制御することを目的に、照合の適合化オプションを指定することもできます。

**参照**


- 「SQL Anywhere データベースの作成」113 ページ
- 「名前を付けた照合データベースの作成」503 ページ
- 「推奨文字セットと照合」506 ページ
- 「CREATE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「初期化ユーティリティ (dbinit)」958 ページ


## 照合の選択時の考慮事項

使用するデータベース用の照合を選択するには、以下について考慮します。

- 文字セット変換を使用すると、パフォーマンスが犠牲になり、またシステム設定が複雑になります。このため、文字セット変換の必要がない照合を選択してください。データベースサーバーとクライアントが同じ文字セットを使用している場合は、文字セット変換は使用されません。

  文字セット変換を回避するには、クライアントコンピューターのオペレーティングシステムで使用されている文字セットに対応する照合順をデータベースで使用します。クライアントコンピューターのオペレーティングシステムが Windows の場合は、ANSI 文字セットを選択してください。

- クライアントコンピューターでさまざまな文字セットを扱う場合、またはデータベースに Unicode データを格納する必要がある場合、UCA 照合か UTF8BIN 照合またはその両方を使用することを検討します。ただし、UTF-8 以外のマルチバイト文字セットに対しては UCA 照合を使用する必要があることに注意してください。

- データベースのデータに適した文字セットとソート順を使用する照合を選択します。この条件を満たす照合が複数ある場合もあります。

## 照合の適合化オプション

データベースの作成時に UCA 照合を選択すると、オプションで照合の適合化オプションを指定できます。UCA 照合を選択しない場合は、適合化構文を使用して大文字と小文字の区別を制御できます。また、COMPARE 関数や SORTKEY 関数を使用してデータの比較やソートを行う場合も適合化オプションを指定できます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### スウェーデン語の照合の適合化

V と W の文字をレベル 1 で区別する Swedish Academy の 2005 年度標準に準拠するよう UCA 照合を適合化するには、UCA (locale=swe;sorttype=phonebook) を指定します。sorttype=phonebook を指定しないと、スウェーデン語のロケールでは V と W は同じ文字として認識されます。

### 日本語の照合の適合化

UCA では、一部のひらがなとカタカナの文字はレベル 3 でのみ区別され、大文字と小文字が区別されない照合ではそれらの違いはなくなります。すべてのひらがな文字間におけるレベル 1 の違いとすべてのカタカナ文字におけるレベル 1 の違いを定義するように UCA 照合を適合化するには、UCA (locale=ja;sorttype=direct;...) を指定します。これらの適合化オプションによって完全に正しいソートセマンティックは提供されませんが、正しい等号セマンティックが提供されます。

### 照合の適合化の概要

照合の適合化オプションは、キーワードと値の組み合わせの形式で指定します。次の表は、サポートされているキーワードを示したものです。使用できる代替形式と値も示します。

**注意**
照合の適合化オプションを使用して作成したデータベースは、10.0.1 より前のデータベースサーバーでは起動できません。

| キーワード | 照合 | 代替形式 | 指定可能な値 |
|---|---|---|---|
| Locale | UCA | (なし) | 任意の有効なロケールコードです。たとえば、en があります。 |

| キーワード | 照合 | 代替形式 | 指定可能な値 |
|---|---|---|---|
| CaseSensitivity | サポートされているすべての照合 | CaseSensitive、Case | ● **respect**　文字の大文字小文字の違いが考慮されます。UCA 照合の場合、UpperFirst を指定したことに等しくなります。その他の照合の場合は、照合によって異なります。<br>● **ignore**　文字の大文字小文字の違いは無視されます。<br>● **UpperFirst**　常に、大文字を先にソートします (Aa)。<br>● **LowerFirst**　常に、小文字を先にソートします (aA)。 |
| AccentSensitivity | UCA | AccentSensitive、Accent | ● **respect**　文字のアクセント記号の違いが考慮されます。<br>● **ignore**　文字のアクセント記号の違いは無視されます。<br>● **French**　フランス語の規則に従ってアクセント記号の違いが考慮されます。 |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| キーワード | 照合 | 代替形式 | 指定可能な値 |
|---|---|---|---|
| PunctuationSensitivity | UCA | PunctuationSensitive、Punct | ● **ignore**　句読表記の違いは無視されます。<br>● **primary**　第 1 レベルのソートを使用します (文字のみを考慮します)。たとえば、a > b になります。<br>● **quaternary**　第 4 レベルのソートを使用します。文字、大文字小文字、アクセント記号、句読表記の順に考慮されます。たとえば、multiByte、multibyte、multi-byte、multi-Byte をソートすると、次のようになります。<br>○ multiByte<br>○ multibyte<br>○ multi-Byte<br>○ multi-byte<br>大文字と小文字やアクセント記号を区別しないデータベースでは、quaternary (第 4 レベル) は指定できません。 |

| キーワード | 照合 | 代替形式 | 指定可能な値 |
|---|---|---|---|
| SortType | UCA | (なし) | 使用するソートタイプです。使用できる値は、次のとおりです。<br>● phonebook<br>● traditional<br>● standard<br>● pinyin<br>● stroke<br>● direct<br>● posix<br>● big5han<br>● gb2312han<br>これらのソートタイプの詳細については、次の URL を参照してください。 |

**参照**

- 「CREATE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「初期化ユーティリティ (dbinit)」958 ページ
- 「COMPARE 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』
- 「SORTKEY 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』

## SQL Anywhere による新しいデータベースのデフォルトの照合の選択方法

新しいデータベースが作成され、照合が明示的に指定されていない場合、SQL Anywhere では言語と文字セットを使用して照合を決定します。

- 言語は、SALANG 環境変数 (存在する場合)、レジストリ、オペレーティングシステムで判断します。
- 文字セットは、SACHARSET 環境変数 (存在する場合) またはオペレーティングシステムで判断します。

**参照**

- 「SALANG 環境変数」450 ページ
- 「SACHARSET 環境変数」449 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

# 国際言語と文字セットのタスク

この項では、国際言語と文字セットの問題に関連したタスクについて、まとめて説明します。

## デフォルトの照合の判断

**♦ 使用コンピューターのデフォルトの照合を判断するには、次の手順に従います。**

データベースの作成時に照合を指定しないと、デフォルトの照合が使用されます。デフォルトの照合は、使用するオペレーティングシステムによって異なります。

1. Interactive SQL を起動します。
2. サンプルデータベースに接続します。
3. 次のクエリを実行します。

   ```
   SELECT PROPERTY( 'DefaultCollation' );
   ```

   デフォルトの照合が返されます。

   この照合の詳細については、「照合の考慮事項」496 ページを参照してください。

## ロケール情報

ロケール情報は、PROPERTY、DB_PROPERTY、CONNECTION_PROPERTY などの関数を使用して確認できます。次の表は、これらの関数を使用して、クライアント接続、データベース、データベースサーバーのロケール情報を返す方法を示します。

| システム関数とパラメーター | 戻り値 |
|---|---|
| SELECT PROPERTY( 'CharSet' ); | データベースサーバーの文字セット。通常は、データベースサーバーを実行しているコンピューターの文字セット |
| SELECT PROPERTY( 'DefaultCollation' ); | データベースサーバーがデータベース作成時に使用するデフォルトの CHAR 照合 |
| SELECT PROPERTY( 'DefaultNcharCollation' ); | データベースサーバーがデータベース作成時に使用するデフォルトの NCHAR 照合 |
| SELECT PROPERTY( 'Language' ); | データベースサーバーのロケール言語 |

| システム関数とパラメーター | 戻り値 |
|---|---|
| SELECT DB_PROPERTY( 'CharSet' ); | CHAR データをデータベースに格納するときに使用する文字セット |
| SELECT DB_PROPERTY( 'NcharCharSet' ); | NCHAR データをデータベースに格納するときに使用する文字セット |
| SELECT DB_PROPERTY( 'MultiByteCharSet' ); | CHAR データでマルチバイト文字セットを使用するかどうか (On=使用、Off=使用しない) |
| SELECT DB_PROPERTY( 'Language' ); | データベースの CHAR 照合でサポートしている言語を表す 2 文字のコードをカンマで区切ったリスト |
| SELECT DB_PROPERTY( 'Collation' ); | データベースサーバーが使用する CHAR 照合名 |
| SELECT DB_PROPERTY( 'NcharCollation' ); | データベースサーバーが使用する NCHAR 照合名 |
| SELECT CONNECTION_PROPERTY( 'CharSet' ); | クライアントの CHAR データ文字セット |
| SELECT CONNECTION_PROPERTY( 'NcharCharSet' ); | 接続に使用する NCHAR データの文字セット |
| SELECT CONNECTION_PROPERTY( 'Language' ); | クライアントが接続に使用する言語 |

**参照**

- 「PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』
- 「DB_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』
- 「CONNECTION_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』

## ロケールの設定

**♦ SQL Anywhere のロケールを設定するには、次の手順に従います。**

1. デフォルトのロケールで問題ない場合は、何の作業も必要ありません。

   オペレーティングシステムのデフォルトロケールを確認する方法については、「ロケール情報」501 ページを参照してください。

2. ロケールを変更する必要がある場合は、SALANG 環境変数か SACHARSET 環境変数のまたはその両方を設定します。

   ```
   SACHARSET=charset SALANG=language-code
   ```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

*charset* は有効な文字セットラベルで、*language-code* は有効な言語のリストにある言語コードです。「言語ラベルの値」485 ページを参照してください。

各種オペレーティングシステムで環境変数を設定する方法については、「SQL Anywhere の環境変数」443 ページを参照してください。

## 名前を付けた照合データベースの作成

データベースの作成時に、各データベースで使用する照合を指定できます。デフォルトの照合は、データベースサーバーのオペレーティングシステムで使用されるコードページと言語から推定されます。

NCHAR 照合の使用方法の詳細については、「NCHAR 照合」491 ページを参照してください。

**♦ データベース作成時にデータベース照合を指定するには、次の手順に従います (コマンドラインを使用する場合)。**

1. 次のコマンドを実行して、推奨する照合順を表示します。

   ```
   dbinit -l
   ```

   リストの最初のカラムは照合ラベルです。これは、データベースの作成時に指定します。

2. dbinit ユーティリティを使用して -z オプションで照合順を指定し、データベースを作成します。次のコマンドは、ギリシャ語の照合を設定したデータベースを作成します。

   ```
   dbinit -z 1253ELL mydb.db
   ```

   次のコマンドを実行すると、大文字と小文字を区別するデータベースである *spanish.db* が作成されます。このデータベースでは、非 NCHAR データに対して 1262spa 照合が使用されます。NCHAR データには、UCA 照合が指定されます。この場合、ロケールは es であり、最初に小文字がソートされます。

   ```
   dbinit -c -z 1252spa -zn uca(locale=es;case=LowerFirst) spanish.db
   ```

**♦ データベース作成時にデータベース照合を指定するには、次の手順に従います (SQL の場合)。**

- CREATE DATABASE 文を使用して、データベースを作成できます。次の文は、ギリシャ語の照合を設定したデータベースを作成します。

  ```
  CREATE DATABASE 'mydb.db' COLLATION '1253ELL';
  ```

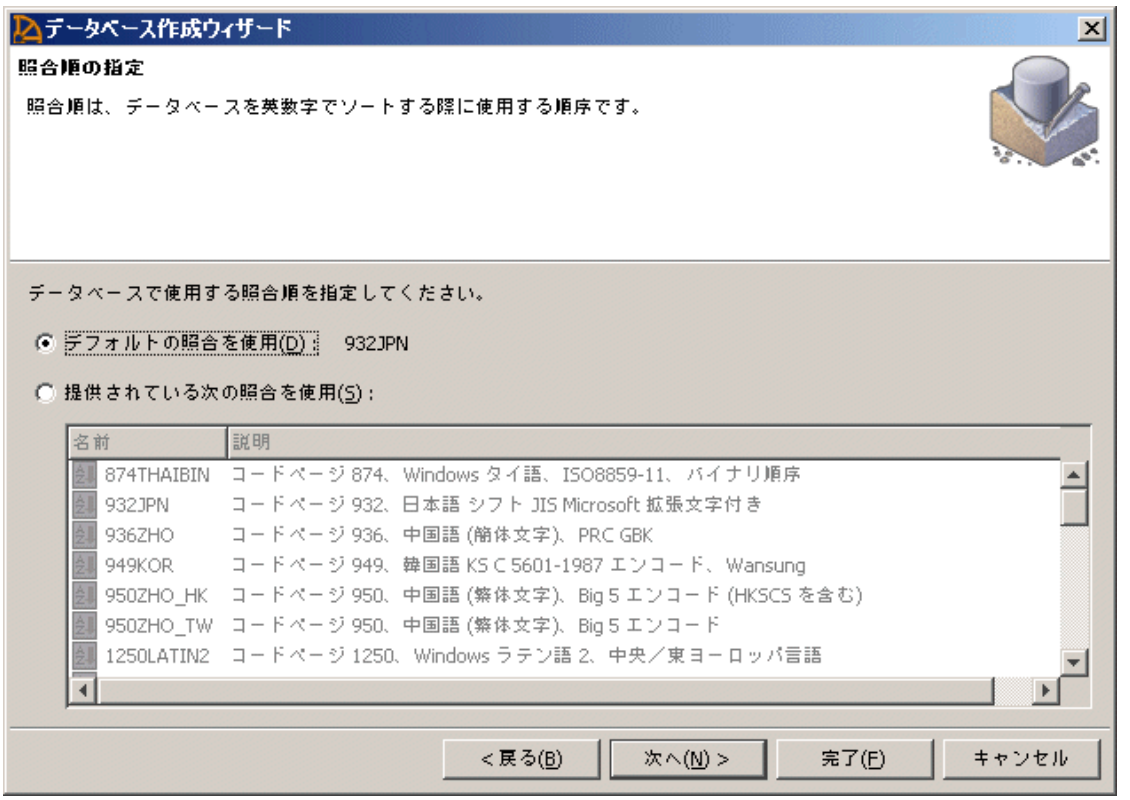
  次の文は、コードページ 1252 を使用してデータベースを作成し、CHAR と NCHAR のデータ型の両方に UCA を使用します。比較とソート時に、アクセント記号と大文字と小文字の区別が考慮されます。

  ```
  CREATE DATABASE 'c:¥¥uca.db' COLLATION 'UCA' ENCODING 'CP1252' NCHAR COLLATION
  'UCA' ACCENT RESPECT CASE RESPECT;
  ```

**♦ データベース作成時にデータベース照合を指定するには、次の手順に従います (Sybase Central の場合)。**

● **[ツール] » [SQL Anywhere 12] » [データベースの作成]** をクリックします。

データベース作成ウィザードの中に、リストから照合を選択するページがあります。




**参照**

- 「チュートリアル：SQL Anywhere データベースの作成」108 ページ
- 「SQL Anywhere データベースの作成」113 ページ
- 「CREATE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「初期化ユーティリティ (dbinit)」958 ページ

## データベースの照合を変更する

**♦ 照合の変更**

データベースの照合を変更するには、データベースを再構築する必要があります。照合はデータベースの作成時に選択されるものであり、変更はできません。

1. データベースを起動します。

```
START databasefile
```

2. 既存のデータベースの文字セットを次のコマンドで確認します。

```
SELECT DB_PROPERTY( 'CharSet' );
```

SQL Anywhere の初期バージョンでは、このプロパティがない場合があります。文字セットは照合名で判断することもできます。たとえば、照合 1252LATIN1 はコードページ 1252 を使用しています。

既存のデータベースの文字セットは、データベースの文字セットと同じか、互換性があるはずです。そうでない場合は、データベースの再構築が必要になりますが、再構築のプロセスには細心の注意が必要です。

特に、使用しているデータベースの照合が 850LATIN1 であり、使用している SQL Anywhere が初期バージョンであるため文字セット変換がサポートされていないか (バージョン 5 以前) デフォルトで無効になっており (バージョン 6 と 7)、クライアントアプリケーションが標準的な Windows アプリケーションの場合、データベースにコードページ 1252 の文字データが含まれている可能性があります (通常はコードページ 850 に含まれる文字データです)。

3. 以下のように、既存のデータベースに格納されているデータの文字セットを確認します。

```
UNLOAD TABLE mytable TO 'mytable_data_in_utf8.dat' ENCODING 'UTF-8';
```

メモ帳で結果を確認します。

アクセント記号付きデータが正しい場合、データベースに含まれている文字データは Windows ANSI コードページに対応しており、英語と西ヨーロッパ言語の場合はコードページ 1252 です。データが DOS ベースのエディターで正常に表示される場合、文字データは Windows OEM コードページに対応しており、通常は 437 または 850 です。

4. 以下のようにデータベースをアンロードします。

データの文字セットがデータベースの文字セットと互換性がない場合、文字セット変換なしでデータをアンロードすることが重要です。使用されている SQL Anywhere によっては、dbunload の内部アンロード機能を使用したり、UNLOAD TABLE 文を使用してデータを手動でアンロードしたりできます。

5. 以下のように新しいデータベースを作成し、使用する照合と文字セットを指定します。

```
dbinit -z 1252LATIN1 c:¥newdatabase.db
```

6. 以下のように古いデータベースサーバーを停止して、新しいデータベースサーバーを起動します。

```
dbeng12 -n new-server c:¥newdatabase.db
```

7. データを新しいデータベースにロードします。

アンロードしたデータとスキーマ (*reload.sql*) が再ロードに使用するコンピューターの文字セットに対応している場合、dbunload の外部再ロードオプションを使用できます。データは、データベースの正しい文字セットに自動的に変換されます。

```
dbunload -ix c:¥databasefile
```

データのエンコードがデータベースの文字セットと一致してなく、データのロードに LOAD TABLE 文 (内部再ロード) を使用している場合は、ENCODING 句を使用する必要があります。データベースサーバーは、LOAD TABLE 文を使用してロードされたデータに対し、デフォルトでは文字セット変換を実行しません。

データのエンコードが作業に使用しているコンピューターのコードページと一致せず、ロードに INPUT 文 (外部再ロード) を使用している場合は、ENCODING 句を使用する必要があります。そうしないと、データベースサーバーはデータがコンピューターのネイティブ文字セットであると想定します。

**参照**

- 「LOAD TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「UNLOAD 文」『SQL Anywhere サーバー SQL リファレンス』
- 「INPUT 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「名前を付けた照合データベースの作成」503 ページ
- 「アンロードユーティリティ (dbunload)」1044 ページ
- 「データベースの再構築」『SQL Anywhere サーバー SQL の使用法』
- 「CharSet (CS) 接続パラメーター」325 ページ

# 文字セットと照合の参考情報

以降の項では、SQL Anywhere の文字セットと照合に関する情報を説明します。

## 推奨文字セットと照合

SQL Anywhere は数多くの文字セット、コードページ、エンコード、照合の名前を認識しますが、この項では Windows プラットフォームと UNIX プラットフォームで使用が推奨されるものを使用言語別に示します。

dbinit -le オプションを使用すると、SQL Anywhere データベースで使用可能なすべての文字セットラベルのリストを取得できます。dbinit -l オプションを使用すると、SQL Anywhere データベースで使用可能な照合のリストを取得できます。「初期化ユーティリティ (dbinit)」958 ページを参照してください。

文字セットエンコードまたはラベルを指定する必要がある場合は、「文字セットラベル」列の値または dbinit -le によって表示されるいずれかのラベルを使用します。照合を指定する必要がある場合は、「照合」列か「代替照合」列の値または dbinit -l によって表示されるいずれかのラベルを使用します。

**注意**
Mac OS X の場合と次の表に示されていない言語については、UTF-8 エンコードを UCA 照合または UTF8BIN 照合と組み合わせて使用してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### Windows プラットフォーム

| 言語 | Windows コードページ | 文字セットラベル | 照合 | 代替照合 |
|---|---|---|---|---|
| アラビア語 | 1256 | Windows -1256 | 1256ARA | |
| 中央および西ヨーロッパ言語 | 1250 | Windows -1250 | 1250LATIN2 | |
| デンマーク語 | 1252 | Windows -1252 | 1252LATIN1 | |
| オランダ語 | 1252 | Windows -1252 | 1252LATIN1 | |
| 英語 | 1252 | Windows-1252 | 1252LATIN1 | |
| フィンランド語 | 1252 | Windows-1252 | 1252SWEFIN | |
| フランス語 | 1252 | Windows-1252 | 1252LATIN1 | |
| ドイツ語 | 1252 | Windows-1252 | 1252LATIN1 | |
| ギリシャ語 | 1253 | Windows -1253 | 1253ELL | |
| ヘブライ語 | 1253 | Windows -1253 | 1255HEB | |
| イタリア語 | 1252 | Windows-1252 | 1252LATIN1 | |
| 日本語 | 932 | Windows-31J | 932JPN | |
| 韓国語 | 949 | IBM949 | 949KOR | |
| リトアニア語 | 1257 | Windows -1257 | 1257LIT | |
| ノルウェー語 | 1252 | Windows-1252 | 1252NOR | |
| ポーランド語 | 1250 | Windows -1250 | 1250POL | |
| ポルトガル語 | 1252 | Windows-1252 | 1252LATIN1 | |
| ロシア語 | 1251 | Windows -1251 | 1251CYR | |
| 中国語 (簡体文字) | 936 | GBK | 936ZHO | |
| スペイン語 | 1252 | Windows-1252 | 1252SPA | |
| スウェーデン語 | 1252 | Windows-1252 | 1252SWEFIN | |
| タイ語 | 874 | TIS-620 | 874THAIBIN | |

| 言語 | Windows コードページ | 文字セットラベル | 照合 | 代替照合 |
|---|---|---|---|---|
| 中国語 (繁体文字) - 香港 | 950 | Big5-HKSCS | 950ZHO_HK | |
| 中国語 (繁体文字) - 台湾 | 950 | Big5 | 950ZHO_TW | |
| トルコ語 | 1254 | Windows -1254 | 1254TRK | 1254TRKALT |
| ウクライナ語 | 1251 | Windows -1251 | 1251CYR | |
| 西ヨーロッパ言語 | 1252 | Windows-1252 | 1252LATIN1 | |

### UNIX プラットフォーム

| 言語 | 文字セットラベル | 照合 | 代替照合 |
|---|---|---|---|
| アラビア語 | ISO_8859-6:1987 | UCA | |
| 中央および西ヨーロッパ言語 | ISO_8859-2:1987 | UCA | |
| デンマーク語 | ISO-8859-15 | ISO9LATIN1 | ISO1LATIN1 |
| オランダ語 | ISO-8859-15 | ISO9LATIN1 | ISO1LATIN1 |
| 英語 | ISO-8859-15 | ISO9LATIN1 | ISO1LATIN1 |
| フィンランド語 | ISO-8859-15 | ISO9LATIN1 | ISO1LATIN1 |
| フランス語 | ISO-8859-15 | ISO9LATIN1 | ISO1LATIN1 |
| ドイツ語 | ISO-8859-15 | ISO9LATIN1 | ISO1LATIN1 |
| ギリシャ語 | ISO_8859-7:1987 | UCA | |
| ヘブライ語 | ISO_8859-8:1988 | UCA | |
| イタリア語 | ISO-8859-15 | ISO9LATIN1 | ISO1LATIN1 |
| 日本語 | EUC-JP[1] | EUC_JAPAN | |
| 韓国語 | EUC-KR | EUC_KOREA | |
| リトアニア語 | (UTF-8 を使用) | UCA または UTF8BIN | |
| ノルウェー語 | ISO-8859-15 | ISO9LATIN1 | ISO1LATIN1 |
| ポーランド語 | ISO_8859-2:1987 | UCA | |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 言語 | 文字セットラベル | 照合 | 代替照合 |
|---|---|---|---|
| ポルトガル語 | ISO-8859-15 | ISO9LATIN1 | ISO1LATIN1 |
| ロシア語 | ISO_8859-5:1988 | UCA | |
| 中国語 (簡体文字) | GB2312 | EUC_CHINA | |
| スペイン語 | ISO-8859-15 | ISO9LATIN1 | ISO1LATIN1 |
| スウェーデン語 | ISO-8859-15 | ISO9LATIN1 | ISO1LATIN1 |
| タイ語 | (UTF-8 を使用) | UCA または UTF8BIN | |
| 中国語 (繁体文字) - 香港 | Big5-HKSCS | 950ZHO_HK | 950TWN |
| 中国語 (繁体文字) - 台湾 | EUC-TW | EUC_TAIWAN | |
| 中国語 (繁体文字) - 台湾 | Big5 | 950ZHO_TW | |
| トルコ語 | ISO_8859-9:1989 | 920TRK | |
| ウクライナ語 | ISO_8859-5:1988 | UCA | |
| 西ヨーロッパ言語 | ISO-8859-15 | ISO9LATIN1 | ISO1LATIN1 |

[1] EUC-JP は、SQL Anywhere ラベルである Extended_UNIX_Code_Packed_Format_for_Japanese の代替ラベルです。

## トルコ語文字セットと照合

トルコ語には、**I** に相当する文字に 2 通りの表記形式があります。1 つ目の形式は、**I-dot** と呼ばれるもので、次のように表記されます。

```
i, İ
```

2 つ目の形式は、**I-no-dot** と呼ばれるもので、次のように表記されます。

```
ı, I
```

これらの文字は同じ文字の変形として表示される場合でも、トルコ語のアルファベットでは別の文字と見なされます。SQL Anywhere では、トルコ語照合 1254TRK を提供して、このような違いに対処しています。

これらの文字の大文字と小文字の変換に関するトルコ語の規則は、ANSI SQL 標準規則と互換性がありません。たとえば、トルコ語では I の小文字に相当するには、以下の文字です。

```
ı
```

一方、ANSI 標準ではこれは次の文字になります。

```
i
```

このような理由から、大文字と小文字を区別しない場合に正しく一致させることができるのは、一致させるテキストがトルコ語か英語／ANSI であるかどうかによります。多くのコンテキストでは、これを区別するだけの十分な情報がないので、そのようなデータベースでは標準外の動作となることがあります。

たとえば、次の文を 1254TRK 照合を使用するデータベースに対して実行するとします。

```
SELECT * FROM syshistory   //actual table name is SYSHISTORY
SELECT * FROM fig          //actual name is FİG
```

最初の文はシステムオブジェクトを参照しており、名前の照合には ANSI SQL 変換規則が必要です。2 番目の文はユーザーオブジェクトを参照しており、名前の照合にはトルコ語変換規則が必要です。ここで、データベースサーバーは、オブジェクトが何であるかを確認できるまでは使用すべき変換規則を判断できず、使用する変換規則が確認できるまではオブジェクトが何であるかを判断できません。この状況においては、システムおよびユーザーオブジェクトの両方を満足させるような解決策はありません。この例では、データベースサーバーがトルコ語照合 1254TRK を使用しており、I の小文字が I の大文字と同じであるとは見なされないので、最初の文は失敗します。2 番目の文は成功します。

トルコ語と ANSI 標準には互換性がないため、トルコ語のデータベースに含まれるシステムオブジェクトを参照する場合は、オブジェクト名の大文字と小文字 (そのオブジェクトの作成時に使用された大文字と小文字) を正確に指定する必要があります。上記の最初の文は次のように記述します。

```
SELECT * FROM SYSHISTORY
```

実際は、文字 I のみ大文字と小文字を正しく対応させます。

別の方法として、通常のやり方ではありませんが、次のように文を記述する方法も可能です。

```
SELECT * FROM syshıstory   //I-no-dot
```

INSERT などのキーワードは、トルコ語データベースでも大文字小文字の区別がありません。SQL Anywhere では、すべてのキーワードが英語の文字のみを使用していると認識しているので、キーワードの一致に ANSI の大文字と小文字の変換規則を使用します。また SQL Anywhere は、組み込み関数などの特定の識別子にこの方法を適用します。ただし、カタログ内に保管されている名前を持つオブジェクトは、上記のように正しい大文字と小文字または文字を使用して指定してください。正しい大文字と小文字を確認するには、システムオブジェクトを定義するシステムビューを調べてください。

**参照**

- 「システムビュー」『SQL Anywhere サーバー SQL リファレンス』

## 大文字と小文字を区別しないトルコ語データベースのデータ

同様の規則で、大文字小文字を区別しないトルコ語データベースのデータを管理します。たとえば、データ値が次のような場合、

```
FİG
```

上記のデータへの小文字参照は、次のようになります。

```
fig
```

同じ I-dot 文字が両方の形式で使用されます。

## 代替トルコ語照合 1254TRKALT

一部のアプリケーション開発者にとって、トルコ語の文字の問題が重大な問題を引き起こす場合もあります。正しい解決策は、すべてのオブジェクト参照先で大文字と小文字が正しく対応していることを確認するか、適切な文字 I を使用しているかを確認することです。ただし、トルコ語の規則に合わせず ANSI 規則を優先した方がうまくいく場合もあります

SQL Anywhere には、照合 1254TRKALT があります。これは、I-dot と I-no-dot を同等の文字とする以外は 1254TRK と同じです。

この変更に伴う違いをきちんと理解しておく必要があります。1254TRKALT データベースでは、次の文字列の区別がありません。

```
fig
fıg
```

これはトルコ語ユーザーにとって正しくはありませんが、他のユーザーにとっては許容範囲である場合もあります。

2 番目の問題は、ORDER BY を使用する場合に出てきます。次の文字列について考えてみます。

```
ia
ıa
ıs
is
```

1254TRK データベースでは、文字列の ORDER BY は次のような順序を生成します。

```
ıa
ıs
ia
is
```

これは、I-no-dot は I-dot より小さいからです。1254TRKALT データベースでは、次のような順序になります。

```
ia
ıa
ıs
is
```

これは、I-no-dot と I-dot が同等だからです。

# ユーザー ID、権限、パーミッション

データベースの各ユーザーは、データベースに接続するときに入力する名前 (ユーザー ID) を持ち、少なくとも 1 つのグループに所属しています。また、ユーザーとグループは、データベース内の情報のセキュリティとプライバシーを維持しながらタスクを実行するために許可された権限とパーミッションを持っています。

「パーミッション」は、テーブル、ビュー、ユーザーなどのデータベースオブジェクトを作成、変更、問い合わせ、使用、削除するために付与されます。「権限」は、データベースのバックアップや診断トレーシングの実行など、データベースレベルでタスクを実行するために付与されます。SQL Anywhere では、パーミッションと権限をユーザーとグループに付与できます。

## ユーザー

ほとんどのデータベースでは、パーミッション管理の大半を個別のユーザーに対してではなく、「グループ」に対して行ってください。ただし、グループは単に特別なプロパティが付加されたユーザー ID にすぎないので、グループ管理に関する説明に移る前に、この項の説明をお読みください。

### 個別のユーザー ID を使用する必要性の理由

マルチユーザーデータベースにおいて、セキュリティが問題にならない場合でも、各ユーザーに対して個別のユーザー ID の設定が必要な場合もあります。パーミッションは、個々のユーザーだけでなく、ユーザーのグループにも付与できます。グループを作成して適切なパーミッションを与えれば、管理作業にかかるオーバーヘッドが非常に少なくなります。

個別のユーザー ID を使用すると、次のような利点があります。

- ログ変換ユーティリティ (dblog) によって、個々のユーザーが加えた変更を必要な部分だけトランザクションログから抽出できます。これはトラブルシューティングのときに非常に有効です。
- Sybase Central では、どの接続がどのユーザーのものであるかわかるように、有用な情報が表示されます。
- ローのロックに関するメッセージ (blocking オプションが Off の場合) の情報量が増えます。

**参照**

- 「グループ」544 ページ

## 新しいユーザーの作成

新しいユーザーはすべて、「PUBLIC」 グループに自動的に追加されます。

DBA 権限を持つユーザーは、特殊な PUBLIC ユーザーグループに対してパーミッションを割り当てることで、新しいユーザーに自動的に与えられるパーミッションを設定できます。「特殊なグループとユーザー」551 ページを参照してください。

デフォルトで新しいユーザーに割り当てられるパーミッションには以下が含まれます。

- データベースへの接続 (ユーザーのパスワードが指定されていることが前提)
- システムビューに格納されているデータの表示(データベースにあるテーブルにアクセスするには、新しいユーザーにパーミッションを付与する必要がある)。
- 大部分のシステムストアドプロシージャーの実行

新しいユーザーを作成すると、次を実行できます。

- そのユーザーを他のグループに追加する。「グループメンバーシップを既存のユーザーまたはグループに付与する」547 ページを参照してください。
- テーブル、ビュー、プロシージャー、シーケンスジェネレーター、DB 領域にそのユーザーのパーミッションを設定する。「ユーザー」512 ページを参照してください。
- そのユーザーにログインポリシーを割り当てる。デフォルトでは、ユーザーはルートログインポリシーに割り当てられます。ただし、カスタムログインポリシーを作成して割り当てることもできます。「ログインポリシー」553 ページを参照してください。
- SQL Remote システムで使用するために、パブリッシャーとしてまたはデータベースのリモートユーザーとしてユーザーを設定する。「ユーザーパーミッション」『SQL Remote』を参照してください。

**参照**

- 「特殊なグループとユーザー」551 ページ
- 「グループの作成」546 ページ

## ユーザーの作成

●ユーザー ID には、次に該当する値を指定できません。

○ 空白スペース、一重引用符、または二重引用符で始まる値
○ 空白スペースで終わる値
○ セミコロンを含む値

●パスワードは大文字と小文字が区別され、次に該当する値は指定できません。

○ 空白スペース、一重引用符、または二重引用符で始まる値
○ 空白スペースで終わる値
○ セミコロンを含む値
○ 256 バイト長を超える値

**♦ 新しいユーザーまたはグループを作成するには、次の手順に従います (Sybase Central の場合)。**

新しいユーザーを作成するには DBA 権限が必要です。

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。

2. 新しいユーザーを作成するには、**[ユーザーとグループ]** を右クリックし、**[新規]** » **[ユーザー]** をクリックします。

3. **ユーザー作成ウィザード**の指示に従います。

**♦ 新しいユーザーを作成するには、次の手順に従います (SQL の場合)。**

新しいユーザーを作成するには DBA 権限が必要です。

1. DBA 権限のあるユーザーとして、データベースに接続します。

2. CREATE USER 文を IDENTIFIED BY を指定して実行します。

**参照**

● 「CREATE USER 文」『SQL Anywhere サーバー SQL リファレンス』
● 「グループの作成」546 ページ

**例**

次の例は、ユーザー ID M_Haneef とパスワード Welcome を使用して、データベースに新しいユーザーを追加します。

```
CREATE USER M_Haneef
IDENTIFIED BY Welcome;
```

# パスワード

ユーザーは、データベースに接続可能なパスワードを持っている必要があります。

作成または変更したパスワードは、UTF-8 に変換されてからハッシュされ、データベースに保存されます。データベースをアンロードし、別の文字セットを使用するデータベースに再ロードした場合でも、既存のパスワードは機能します。サーバーがクライアントの文字セットを UTF-8 に変換できない場合、パスワードには 7 ビット ASCII 文字を使用することをおすすめします。それ以外の文字を使用すると、パスワードが機能しないことがあります。

**警告**
ユーザー ID DBA のデフォルトのパスワードは、すべてのデータベースで sql です。データベースに対する不正なアクセスを防ぐために、このパスワードを変更してください。「パスワードのセキュリティ」1383 ページを参照してください。

- ユーザー ID には、次に該当する値を指定できません。
  - 空白スペース、一重引用符、または二重引用符で始まる値
  - 空白スペースで終わる値
  - セミコロンを含む値
- パスワードは大文字と小文字が区別され、次に該当する値は指定できません。
  - 空白スペース、一重引用符、または二重引用符で始まる値
  - 空白スペースで終わる値
  - セミコロンを含む値
  - 256 バイト長を超える値

**参照**

- 「パスワードのないグループ」550 ページ

# パスワードの変更

**♦ パスワードを変更するには、次の手順に従います (Sybase Central の場合)。**

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。
2. **[ユーザーとグループ]** をクリックします。
3. **[ユーザーとグループ]** リストでユーザーを右クリックし、**[プロパティ]** をクリックします。
4. **[このユーザーにパスワードを設定]** をクリックします。
5. **[パスワード]** フィールドと **[パスワードの確認]** フィールドに値を入力します。
6. **[適用]** をクリックします。

7. **[OK]** をクリックします。

**♦ パスワードを変更するには、次の手順に従います (SQL の場合)。**

1. DBA 権限のあるユーザーとしてデータベースに接続します。

2. ALTER USER 文を実行します。

●ユーザー ID には、次に該当する値を指定できません。

○ 空白スペース、一重引用符、または二重引用符で始まる値
○ 空白スペースで終わる値
○ セミコロンを含む値

●パスワードは大文字と小文字が区別され、次に該当する値は指定できません。

○ 空白スペース、一重引用符、または二重引用符で始まる値
○ 空白スペースで終わる値
○ セミコロンを含む値
○ 256 バイト長を超える値

**参照**

● 「ALTER USER 文」『SQL Anywhere サーバー SQL リファレンス』

**例**

次の文を実行して、DBA ユーザーのパスワードを welcome_DBA に変更します。

```
ALTER USER DBA
IDENTIFIED BY welcome_DBA;
```

## ユーザーとグループのオプションの設定

**♦ ユーザーまたはグループのオプションを設定するには、次の手順に従います (Sybase Central の場合)。**

Sybase Central では、ユーザーやグループのための設定オプションは、**[ユーザーのオプション]** ウィンドウと **[グループのオプション]** ウィンドウ (データベースのオプションを設定するためのウィンドウと同じもの) にあります。Interactive SQL では、SET OPTION 文でオプションを指定できます。

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。

2. **[ユーザーとグループ]** をクリックします。

3. ユーザーまたはグループを右クリックし、**[オプション]** をクリックします。

4. **[オプション]** リストでオプションをクリックします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

5. **[恒久的な設定を行う]** をクリックします。

6. **[閉じる]** をクリックします。

**♦ ユーザーまたはグループのオプションを設定するには、次の手順に従います (SQL の場合)。**

Sybase Central では、ユーザーやグループのための設定オプションは、**[ユーザーのオプション]** ウィンドウと **[グループのオプション]** ウィンドウ (データベースのオプションを設定するためのウィンドウと同じもの) にあります。Interactive SQL では、SET OPTION 文でオプションを指定できます。

1. DBA 権限のあるユーザーとして、データベースに接続します。

2. SET OPTION 文を実行します。

   たとえば、PUBLIC グループ内のすべてのユーザーに対して日付フォーマットオプションを設定するには、次の文を実行します。

```
SET OPTION PUBLIC.date_format = 'Mmm dd yyyy';
```

**参照**

- 「データベースオブジェクトのプロパティの設定」『SQL Anywhere サーバー SQL の使用法』
- 「データベースオプション」575 ページ

## ユーザーをデータベースから削除する

**♦ データベースからユーザーを削除するには、次の手順に従います (Sybase Central の場合)。**

ユーザーを削除すると、そのユーザーが所有するすべてのデータベースオブジェクト (テーブルなど) とそのユーザーの外部ログインも削除されます。

ユーザーの削除中は、対象ユーザーはデータベースに接続できません。

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。

   ユーザーを削除できるのは、DBA 権限を持つユーザーだけです。

2. **[ユーザーとグループ]** をクリックします。

3. ユーザーを右クリックし、**[削除]** をクリックします。

4. **[はい]** をクリックします。

**♦ データベースからユーザーを削除するには、次の手順に従います (SQL の場合)。**

ユーザーを削除すると、そのユーザーが所有するすべてのデータベースオブジェクト (テーブルなど) とそのユーザーの外部ログインも削除されます。

ユーザーの削除中は、対象ユーザーはデータベースに接続できません。

1. DBA 権限のあるユーザーとして、データベースに接続します。

2. DROP USER 文を実行します。

**参照**

- 「DROP USER 文」『SQL Anywhere サーバー SQL リファレンス』
- 「REVOKE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「ユーザーパーミッションの取り消し」532 ページ
- 「グループをデータベースから削除する」552 ページ

**例**

データベースからユーザー M_Haneef を削除します。

```
DROP USER M_Haneef;
```

## システムビューに対するユーザーとパーミッション

データベースのシステムビューには、データベースの現在のユーザーとそのパーミッションに関する情報が含まれています。

システムビューの所有者は、特殊なユーザー SYS です。ユーザー ID SYS を使用して接続することはできません。

DBA 権限を持つユーザーには、すべてのシステムビューにアクセスする SELECT パーミッションはありますが、基本となるシステムテーブルへの SELECT パーミッションはありません。他のユーザーについても、一部のテーブルとビューへのアクセスは制限されます。たとえば、SYS.SYSUSERPERM ビューにアクセスできるのは DBA 権限を持つユーザーだけです。このビューには、各ユーザー ID の暗号化されたパスワードなど、データベースのユーザーのパーミッションに関するすべての情報が含まれています。ただし、SYS.SYSUSERPERMS ビューには、パスワード以外は SYS.SYSUSERPERM と同じ情報が含まれています。デフォルトで、すべてのユーザーが SYS.SYSUSERPERMS への SELECT アクセス権を持っています。

新しいデータベース内で SYS、PUBLIC、DBA、および dbo に対して設定されたすべてのパーミッションとグループメンバーシップは、自由に変更できます。

次の表に、ユーザー ID、グループ、パーミッションに関する情報を含むシステムビューの概要を示します。ユーザー ID SYS はリストされたすべてのビューを所有し、その修飾された名前は SYS.SYSUSERPERM などになります。これらのビューに対して適切な SELECT クエリを使用すると、すべてのユーザー ID とパーミッションの情報を得られます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| ビュー | デフォルト | 内容 |
|---|---|---|
| SYSCOLAUTH | PUBLIC | SYSCOLPERM の情報を読みやすくしたもの。「SYSCOLAUTH 統合ビュー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| SYSCOLPERM | PUBLIC | GRANT 文で与えられる SELECT または UPDATE パーミッションを持つすべてのカラム。「SYSCOLPERM システムビュー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| DUMMY | PUBLIC | 現在のユーザー ID を知るのに使用できるダミーテーブル。「DUMMY システムテーブル」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| SYSGROUP | PUBLIC | 各グループのメンバーごとに 1 ロー追加。「SYSGROUP システムビュー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| SYSGROUPS | PUBLIC | SYSGROUP の情報を読みやすくしたもの。「SYSGROUPS 統合ビュー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| SYSPROCAUTH | PUBLIC | SYSPROCPERM の情報を読みやすくしたもの。「SYSPROCAUTH 統合ビュー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| SYSPROCPERM | PUBLIC | 各ローには、1 つのプロシージャーを使うパーミッションを与えられた 1 人のユーザーが含まれる。「SYSPROCPERM システムビュー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| SYSTABAUTH | PUBLIC | SYSTABLEPERM の情報を読みやすくしたもの。「SYSTABAUTH 統合ビュー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |

| ビュー | デフォルト | 内容 |
|---|---|---|
| SYSTABLEPERM | PUBLIC | GRANT 文で付与されるテーブルに関するすべてのパーミッション。「SYSTABLEPERM システムビュー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| SYSUSER | DBA のみ | データベース内のすべてのユーザーに関する情報。「SYSUSER システムビュー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| SYSUSERAUTH | DBA のみ | ユーザー ID 以外のすべての SYSUSERPERM の情報。「SYSUSERAUTH 互換ビュー (旧式)」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| SYSUSERAUTHORITY | PUBLIC | 各ユーザー ID に対して付与されている権限。「SYSUSERAUTHORITY システムビュー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| SYSUSERLIST | PUBLIC | パスワード以外のすべての SYSUSERAUTH の情報。「SYSUSERLIST 互換ビュー (旧式)」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| SYSUSERPERM | DBA のみ | データベースレベルのパーミッションと各ユーザー ID のパスワード。「SYSUSERPERM 互換ビュー (旧式)」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| SYSUSERPERMS | PUBLIC | パスワード以外のすべての SYSUSERPERM の情報。「SYSUSERPERMS 互換ビュー (旧式)」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |

**参照**

● 「特殊なグループとユーザー」 551 ページ

# パーミッション

ユーザーのパーミッションは、次のメインカテゴリにグループ分けできます。

●「ユーザーまたはグループに対して明示的に付与されるパーミッション」 524 ページ

●「オブジェクトを所有することで取得したパーミッション」525 ページ

●「グループメンバーシップを通して継承したパーミッション」525 ページ

●「無効なオブジェクトに付与されるパーミッション」526 ページ

次の表は、データベースオブジェクトに付与できるパーミッションをまとめたものです。

| パーミッションタイプ | データベースオブジェクトでサポートされるもの | グループメンバーシップを通して継承可能 | 説明 | WITH GRANT オプションサポート |
|---|---|---|---|---|
| ALL | テーブル、ビュー | はい | データベースオブジェクトに関連するすべてのタスクを実行することをユーザーに許可します。このパーミッションは、ALTER、DELETE、INSERT、REFERENCES、SELECT、および UPDATE の各パーミッションをテーブルに付与します。このパーミッションは、DELETE、INSERT、SELECT、および UPDATE の各パーミッションをビューに付与します。「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 | GRANT と WITH GRANT OPTION |
| ALTER | テーブル | はい | テーブルの構造を変更することや、テーブルにトリガーを作成することをユーザーに許可します。このパーミッションは、データベーススキーマを変更するパーミッションをユーザーに付与するものなので、ほとんどのユーザーには付与する必要はありません。「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 | GRANT と WITH GRANT OPTION |
| CONNECT | ユーザー | いいえ | データベースへの接続をユーザーに許可する「新しいユーザーの作成」513 ページを参照してください。 | なし |
| CONSOLIDATE | ユーザー | いいえ | SQL Remote で使用する統合データベースを指定する「CONSOLIDATE パーミッション」『SQL Remote』を参照してください。 | なし |

| パーミッションタイプ | データベースオブジェクトでサポートされるもの | グループメンバーシップを通して継承可能 | 説明 | WITH GRANT オプションサポート |
|---|---|---|---|---|
| CREATE ON | DB 領域 | はい | RESOURCE 権限で、指定された DB 領域にデータベースオブジェクトを作成することをユーザーに許可します。「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』と「リソース権限」541 ページを参照してください。 | GRANT と WITH GRANT OPTION |
| DELETE | テーブル、ビュー | いいえ | テーブルまたはビューからローを削除することをユーザーに許可します。「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 | GRANT と WITH GRANT OPTION |
| EXECUTE | プロシージャー、ユーザー定義関数 | いいえ | プロシージャーまたは関数の実行をユーザーに許可します。「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 | GRANT ただし、WITH GRANT OPTION は指定しない |
| INSERT | テーブル、ビュー | はい | テーブルまたはビューへのローの挿入をユーザーに許可します。「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 | GRANT と WITH GRANT OPTION |
| INTEGRATED LOGIN | ユーザー | いいえ | 統合化ログインを使用したデータベースへの接続をユーザーに許可する「Windows 統合ログイン」51 ページを参照してください。 | なし |
| KERBEROS LOGIN | ユーザー | いいえ | Kerberos ログインを使用したデータベースへの接続をユーザーに許可する「Kerberos 認証」61 ページを参照してください。 | なし |
| PUBLISH | ユーザー | いいえ | SQL Remote でのデータベースのパブリッシャーを指定する「PUBLISH パーミッション」『SQL Remote』を参照してください。 | なし |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| パーミッションタイプ | データベースオブジェクトでサポートされるもの | グループメンバーシップを通して継承可能 | 説明 | WITH GRANT オプションサポート |
|---|---|---|---|---|
| REFERENCES | テーブル | はい | テーブルへのインデックスの作成とテーブルを参照する外部キーの作成をユーザーに許可します。また、このパーミッションは、テーブル内の個別のカラムに対しても付与できます。このパーミッションは、データベーススキーマを変更するパーミッションをユーザーに付与するものなので、ほとんどのユーザーには割り当てる必要はありません。<br><br>カラム名を指定する場合、ユーザーはテーブル内のそれらのカラムのみを参照できます。「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 | GRANT と WITH GRANT OPTION |
| REMOTE | ユーザー | いいえ | SQL Remote と Mobile Link で使用するリモートデータベースを指定する「GRANT REMOTE DBA 文 [Mobile Link] [SQL Remote]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 | なし |
| SELECT | テーブル、ビュー | はい | テーブルまたはビューの情報を見ることをユーザーに許可します。また、このパーミッションは、テーブル内の個別のカラムに対しても付与できます。「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 | GRANT と WITH GRANT OPTION |
| UPDATE | テーブル、ビュー | はい | テーブルまたはビューのローを更新することをユーザーに許可します。また、このパーミッションは、テーブル内の個別のカラムに対しても付与できます。「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 | GRANT と WITH GRANT OPTION |

| パーミッションタイプ | データベースオブジェクトでサポートされるもの | グループメンバーシップを通して継承可能 | 説明 | WITH GRANT オプションサポート |
|---|---|---|---|---|
| USAGE | シーケンスジェネレーター | いいえ | シーケンス内の現在の値または次の値を評価することをユーザーに許可します。このパーミッションを使用するために、ユーザーは DBA 権限を持つか、シーケンスの作成者である必要があります。シーケンスがテーブルの DEFAULT 句の一部である場合、テーブルにローを挿入するすべてのユーザーにシーケンスに対するパーミッションが必要です。「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 | GRANT ただし、WITH GRANT OPTION は指定しない |
| WITH GRANT OPTION | テーブル、ビュー | いいえ | 同じパーミッションを他のユーザーに付与することをユーザーに許可します。 | |

## ユーザーまたはグループに対して明示的に付与されるパーミッション

これらは、ユーザーまたはグループが指定のデータベースオブジェクトを作成、実行、削除、変更、使用できるかどうかを制御するために明示的に設定されたパーミッションです。パーミッションは、GRANT 文を使用するか、Sybase Central でデータベースオブジェクトの **[パーミッション]** タブを使用して明示的に設定されます。「ユーザーへのパーミッションの付与」528 ページを参照してください。

### ネガティブパーミッション

SQL Anywhere はネガティブパーミッションをサポートしません。つまり、明示的に付与されていないパーミッションは取り消せません。

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

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**参照**

- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「REVOKE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「パーミッション」520 ページ
- 「ユーザーへのパーミッションの付与」528 ページ

## オブジェクトを所有することで取得したパーミッション

データベース内に新しくオブジェクトを作成したユーザーは、そのオブジェクトの「所有者」と呼ばれます。所有者には、そのオブジェクトに対してすべての操作を行うパーミッションが自動的に与えられます。たとえば、テーブルの所有者はそのテーブルの構造を変更したり、他のユーザーにテーブルのデータを更新するパーミッションを与えたりすることができます。

DBA 権限を持つユーザーは、次のパーミッションを持っています。

- 他のユーザーが作成したオブジェクトを含め、データベース内のすべてのコンポーネントを変更すること。たとえば、DBA 権限を持つユーザーは、別のユーザーが作成したテーブルを削除できます。DBA 権限を持つユーザーは、各データベースオブジェクトの所有者がそのオブジェクトに関して持っているパーミッションをすべて持っています。

- 他のユーザーのためにデータベースオブジェクトを作成すること。この場合、オブジェクトの所有者と CREATE 文を実行するユーザー ID が異なります。このマニュアルでは、データベースオブジェクトに関しては、所有者と作成者という用語を同一のものとして使用します。

**参照**

- 「パーミッション」520 ページ
- 「ユーザーへのパーミッションの付与」528 ページ

## グループメンバーシップを通して継承したパーミッション

各ユーザーに対して個別にパーミッションを設定するのは時間がかかり、またエラーの原因になります。ほとんどのデータベースでは、グループ単位のパーミッション管理の方が、個々のユーザー ID 単位の管理よりはるかに効率的です。

各ユーザー ID は複数グループのメンバーとなることができ、各グループの継承可能なパーミッションを継承します。たとえば、会社のデータベース内で、部署ごとに (営業部やマーケティング部などの) グループを作成し、それらのグループにパーミッションを与えることができます。各営業部員は営業部グループのメンバーになり、自動的にデータベースの適切な領域へアクセスできるようになります。

グループのメンバーは、属しているグループの次のパーミッションだけを継承できます。

- ALL
- ALTER
- DELETE
- INSERT
- REFERENCES
- SELECT
- UPDATE

テーブルとビューについては、同じパーミッションを他のユーザーに付与するために、ユーザーやグループにパーミッションを付与できます。同じパーミッションを他のユーザーに付与するこのパーミッションは、グループのメンバーによっては継承されません。

**参照**


- 「パーミッションを他のユーザーに付与する権利をユーザーに付与する」531 ページ
- 「グループ」544 ページ
- 「パーミッション」520 ページ
- 「パスワードのないグループ」550 ページ
- 「ユーザーへのパーミッションの付与」528 ページ


## 無効なオブジェクトに付与されるパーミッション

パーミッションは無効なオブジェクトに対して付与できます。無効なオブジェクトに対するパーミッションはデータベースに保存され、オブジェクトが有効になると使用できるようになります。

**参照**


- 「パーミッション」520 ページ
- 「ユーザーへのパーミッションの付与」528 ページ


## データベースオブジェクトパーミッション

次のデータベースオブジェクトに対して、パーミッションを付与できます。

### テーブルパーミッション

DBA 権限を持つユーザーまたはテーブルの所有者は、テーブル、またはテーブルの個別のカラムに対するパーミッションを付与できます。テーブルに付与できるパーミッションのリストについては、「パーミッション」520 ページを参照してください。

テーブルへのパーミッションは、GRANT 文を使用するか、Sybase Central でデータベースオブジェクトの **[パーミッション]** タブを使用することで付与できます。「ユーザーへのパーミッションの付与」528 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### ビューパーミッション

DBA 権限を持つユーザーまたはビューの所有者は、そのビューに対するパーミッションを付与できます。ビューに付与できるパーミッションのリストについては、「パーミッション」520 ページを参照してください。

次の 1 つ以上の項目が当てはまる場合、ユーザーはビューに対する DELETE、INSERT、SELECT、UPDATE パーミッションを取得できます。

- DBA 権限を持つユーザー、またはパーミッションを明示的に付与されたビューの所有者。
- ビュー内で使用されるすべての基本オブジェクト (ベーステーブル、ビュー、プロシージャーなど) の操作に対する適切なパーミッションを保持するユーザー。
- ビューに対する特定の操作のパーミッションが、DBA 以外のユーザーによって正しく付与された場合。このユーザーは、ビューの所有者であるか、またはビューに対する適切なパーミッション WITH GRANT OPTION を所有する必要があります。ビューの所有者は次のいずれかです。
  - DBA 権限を持つユーザー。
  - DBA 権限を持っていないが、ビューによって参照されるすべてのベーステーブルの所有者であるユーザー。
  - DBA 権限を持っていないユーザーで、ビューによって参照される一部またはすべてのベーステーブルの所有者ではない者。ただし、ビューの所有者は、所有していないベーステーブルに WITH GRANT OPTION で SELECT パーミッションを持ち、所有していないベーステーブルに WITH GRANT OPTION で操作に対するその他の必要なパーミッションを持っています。

    所有者がベーステーブルに対するパーミッション (WITH GRANT OPTION) を保持する代わりに、システムテーブルの SELECT パーミッションを含む PUBLIC にパーミッションが付与される場合もあります。

> **注意**
> パーミッションは無効なビューに対して付与できます。無効なビューに対するパーミッションはデータベースに保存され、ビューが有効になると使用できるようになります。

「システムビューに対するユーザーとパーミッション」518 ページを参照してください。

### プロシージャパーミッション

DBA 権限を持つユーザーまたはプロシージャーの所有者は、ストアドプロシージャーに対するパーミッションを付与できます。EXECUTE パーミッションは、プロシージャーに対して付与できる唯一のパーミッションです。

**プロシージャーの実行パーミッション**　プロシージャーは所有者のパーミッションのもとで実行されます。テーブルの情報を更新するプロシージャーが正しく実行されるのは、そのプロシージャーの所有者が対象となるテーブルの UPDATE パーミッションを持っている場合のみです。

プロシージャーの所有者が適切なパーミッションを持っていれば、そのプロシージャーの EXECUTE パーミッションを割り当てられているユーザーは、基本となるテーブルに対するパー

ミッションの有無にかかわらず、そのプロシージャーを実行できます。プロシージャーを使用すると、テーブルに対する一般的なパーミッションがなくても、ユーザーがテーブルに一定の作業を行う許可を与えることができます。

### トリガーパーミッション

DBA 権限を持つユーザーまたはトリガーの所有者は、トリガーを実行するパーミッションを付与できます。EXECUTE パーミッションは、トリガーに対して付与できる唯一のパーミッションです。

ユーザーのアクションに応じて、データベースサーバーがトリガーを実行します。EXECUTE パーミッションがトリガーに付与されていると、トリガーは実行にさらなるパーミッションを必要としません。トリガーの実行は、関連するテーブルの作成者のパーミッションで行われます。

### DB 領域パーミッション

DBA 権限を持つユーザーは、指定された DB 領域にオブジェクトを作成するパーミッションを付与できます。DB 領域へのパーミッションを付与されたユーザーは、RESOURCE 権限を持つ必要があります。CREATE ON パーミッションは、DB 領域に対して付与できる唯一のパーミッションです。

### シーケンスジェネレーターパーミッション

DBA 権限を持つユーザーまたはシーケンスの作成者は、シーケンスを実行するパーミッションを付与できます。USAGE パーミッションは、シーケンスジェネレーターに付与される唯一のパーミッションです。このパーミッションは、ユーザーがシーケンス内の現在の値または次の値を評価できるようにします。シーケンスがテーブルの DEFAULT 句の一部である場合、テーブルにローを挿入するすべてのユーザーにシーケンスに対するパーミッションが必要です。

### 参照


- 「パーミッション」520 ページ
- 「ユーザーへのパーミッションの付与」528 ページ
- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「トリガーを実行するパーミッション」『SQL Anywhere サーバー SQL の使用法』
- 「リソース権限」541 ページ
- 「ユニークな値を生成するためのシーケンスの使用」『SQL Anywhere サーバー SQL の使用法』


## ユーザーへのパーミッションの付与

データベースオブジェクトにユーザーパーミッションを付与するには、GRANT 文または Sybase Central のいずれかを使用します。

Sybase Central では、ユーザーごとまたはデータベースオブジェクトごとでパーミッションを表示または変更できます。たとえば、特定のテーブルに対するパーミッションを誰が持っているのかを確認する場合、左側のウィンドウ枠にあるテーブルをクリックし、右側のウィンドウ枠にあるその **[パーミッション]** タブをクリックすることによって、テーブルのパーミッションを表示できます。同様に、特定のユーザーに対するすべてのパーミッションを変更する場合、左側のウィンドウ枠にあるユーザーをクリックし、右側のウィンドウ枠でその **[テーブルのパーミッ**

ション] タブをクリックすることによって、すべてのユーザーの明示的に設定されたテーブルパーミッションを表示できます。

**♦ テーブル、ビュー、カラム、プロシージャー、DB 領域、シーケンスジェネレーターでユーザーにパーミッションを付与するには、次の手順に従います (Sybase Central の場合)。**

1. **[SQL Anywhere 12]** プラグインを使用して、DBA 権限を持つユーザーとしてまたはパーミッションを付与したいデータベースオブジェクトを所有しているユーザーとしてデータベースに接続します。

2. 左側のウィンドウ枠で、ユーザーパーミッションを付与したいデータベースオブジェクトをクリックするか、パーミッションを付与したいユーザー (またはグループ) をクリックします。

   - **ユーザーごとにパーミッションを付与するには**　ユーザー (またはグループ) をクリックした場合、右側のウィンドウ枠で、パーミッションを付与したいデータベースオブジェクトに対応するパーミッションタブをクリックします。たとえば、テーブルにパーミッションを付与するには、**[テーブルのパーミッション]** タブをクリックします。

     パーミッションタブには、ユーザー (またはグループ) が明示的にパーミッションを設定しているデータベースオブジェクト名のリストが表示されています。データベースオブジェクトに関するパーミッションをユーザーに付与するには、右クリックして、**[新規]** » **[パーミッション]** をクリックします。

   - **データベースオブジェクトごとにパーミッションを付与するには**　データベースオブジェクトをクリックした場合、右側のウィンドウ枠で **[パーミッション]** タブをクリックします。

     **[パーミッション]** タブでは、**[被付与者]** カラムに、データベースオブジェクトに現在パーミッションを持っているユーザーとグループのリストが表示されます。ユーザーとグループに対してパーミッションを付与するには、右クリックして、**[新規]** » **[パーミッション]** をクリックします。

3. データベースオブジェクトまたは変更しようとしているパーミッションを持つユーザーのローを選択します。

   - **DB 領域、プロシージャー、関数、シーケンスジェネレーター、ビューに対してパーミッションを付与するには**　データベースオブジェクトの横にあるパーミッションカラムをクリックすると、チェックマークが追加されます。チェックマークによって、ユーザーまたはグループにパーミッションが付与されます。

   - **テーブルおよびテーブルカラムにパーミッションを付与するには**　テーブルを選択するときに、対応する **[カラムのパーミッション]** テーブルが **[パーミッション]** テーブルの下に表示されます。

     **テーブルにパーミッションを付与するには**　**[テーブルのパーミッション]** テーブルでは、ユーザーまたはグループの横にあるパーミッションカラムをクリックし、チェックマークまたはプラス記号 (+) 付きのチェックマークを追加します。

     チェックマークによって、ユーザー (またはグループ) にパーミッションが付与されます。プラス記号 (+) 付きのチェックマークは、ユーザーにパーミッションを付与し、ユーザーに他のユーザーへの同じパーミッションを付与するパーミッションを付与します。プラ

ス記号 (+) 付きのチェックマークは、WITH GRANT OPTION 句付きの GRANT 文を実行することと同等です。

**テーブルカラムにパーミッションを付与するには** **[カラムのパーミッション]** テーブルでは、テーブルカラムの横にあるカラムをクリックし、チェックマークまたはプラス記号 (+) 付きのチェックマークを追加します。

**[カラムのパーミッション]** テーブルでカラムを変更するときに、**[付与されている、または付与オプション付きで付与されているカラムパーミッション]** アイコンが前述の **[テーブルのパーミッション]** に表示されます。

パーミッションをグループに付与するときに、パーミッションはグループのメンバーによって継承されます。同じパーミッションを他のユーザーに付与するためのこのパーミッションは、グループのメンバーには継承されません。

4. すべての変更を保存するには、**[ファイル]** » **[保存]** をクリックします。

**ヒント**

- 変更を取り消すには、パーミッションタブにある 1 つ以上のローを選択して、**[編集]** » **[元に戻す]** をクリックします。
- パーミッションのテーブルに対して **[凡例]** をオンまたはオフにするには、**[ファイル]** » **[凡例を表示]** をクリックします。

**♦ テーブル、ビュー、カラム、プロシージャー、DB 領域、シーケンスジェネレーターに対してパーミッションを付与するには、次の手順に従います (SQL の場合)。**

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. GRANT 文を実行してパーミッションを割り当てます。

**参照**

- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「パーミッションの取り消し」533 ページ

**例**

1. テーブルパーミッションはすべて、よく似た方法で付与されます。たとえば、次の手順では、M_Haneef に sample_table というテーブルからローを削除するパーミッションを与えることができます。

   a. DBA 権限のあるユーザーまたは sample_table の所有者としてデータベースに接続します。

   b. 次の SQL 文を実行します。

```
GRANT DELETE
ON sample_table
TO M_Haneef;
```

2. 次の手順で、sample_table という名前のテーブルに含まれるカラム column_1 と column_2 だけを更新するパーミッションを M_Haneef に与えることができます。

   a. DBA 権限のあるユーザーまたは sample_table の所有者としてデータベースに接続します。

   b. 次の SQL 文を実行します。

```
GRANT UPDATE ( column_1, column_2 )
ON sample_table
TO M_Haneef;
```

   テーブルパーミッションの対象は、テーブル内の全データに制限されます。ただし、REFERENCES、SELECT、および UPDATE パーミッションはカラムのサブセットに付与できます。さらに細かいユーザーパーミッションを設定するには、テーブルに対してアクションを実行するプロシージャーを作成し、そのプロシージャーを実行するパーミッションをユーザーに与えます。

3. 次の手順では、プロシージャー my_procedure を実行するパーミッションを M_Haneef に与えることができます。

   a. DBA 権限のあるユーザーまたは my_procedure プロシージャーの所有者としてデータベースに接続します。

   b. 次の SQL 文を入力して実行します。

```
GRANT EXECUTE
ON my_procedure
TO M_Haneef;
```

## パーミッションを他のユーザーに付与する権利をユーザーに付与する

テーブルとビューについては、同じパーミッションを他のユーザーに付与するために、ユーザーにパーミッションを付与できます。GRANT 文で WITH GRANT OPTION を使用して、このパーミッションをユーザーに付与します。同じパーミッションを他のユーザーに付与するこのパーミッションは、グループのメンバーによっては継承されません。

グループのメンバーは、属しているグループの次のパーミッションだけを継承できます。

- ALL
- ALTER
- DELETE
- INSERT
- REFERENCES
- SELECT
- UPDATE

**♦ テーブルとビューにパーミッションを付与して、同じパーミッションを他のユーザーに付与するには、次の手順に従います (SQL の場合)。**

1. DBA 権限のあるユーザーとして、またはテーブルまたはビューの所有者のユーザーとして、データベースに接続します。

2. GRANT 文を WITH GRANT OPTION 付きで実行することでパーミッションを割り当てます。「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

   次に例を示します。

```
GRANT DELETE ON sample_table
TO M_Haneef
WITH GRANT OPTION;
```

**♦ テーブルとビューにパーミッションを付与して、同じパーミッションを他のユーザーに付与するには、次の手順に従います (Sybase Central の場合)。**

- 「ユーザーへのパーミッションの付与」528 ページを参照してください。

**参照**

- 「ユーザーパーミッションの取り消し」532 ページ
- 「テーブルパーミッション」526 ページ
- 「グループのパーミッションと権限」549 ページ

## ユーザーパーミッションの取り消し

ユーザーのパーミッションは、付与されたパーミッションと取り消されたパーミッションの組み合わせで決まります。パーミッションの付与と取り消しを使って、データベースのユーザーパーミッションを管理できます。データベースオブジェクトに対してパーミッションを付与または取り消すには、DBA 権限のあるユーザーか、データベースオブジェクトの所有者である必要があります。

REVOKE 文を実行すると、ユーザーに明示的に付与されたパーミッション (ユーザーが所属するグループから継承されたものではないパーミッション) が取り消されます。

### グループパーミッションの取り消し

ユーザーをグループに追加するとき、ユーザーはパーミッション (WITH GRANT OPTION を使用して同じパーミッションをユーザーに付与するパーミッションを除く) とそのグループに割り当てられる継承可能な権限を継承します。SQL Anywhere では、グループのメンバーとしてユーザーが継承するパーミッションと権限のサブセットを取り消すことはできません。取り消しが可能なパーミッションは、GRANT 文によって明示的に付与されたパーミッションだけです。継承されたパーミッションまたは権限をユーザーから削除する必要がある場合は、新しいグループを作成して必要なパーミッションと権限を割り当てた上でユーザーをそのグループのメンバーにするか、ユーザーをグループから削除して必要なパーミッションを明示的に付与します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### WITH GRANT OPTION の取り消し

ユーザーへのパーミッションを取り消し、またパーミッションに WITH GRANT OPTION があった場合、そのユーザーがパーミッションを付与していたすべてのユーザーへのパーミッションも取り消されます。たとえば、UserA SELECT...WITH GRANT OPTION パーミッションをテーブルと UserA に付与し、SELECT パーミッションをテーブルで UserB に付与します。UserA から SELECT パーミッションを取り消すと、UserB でも取り消されます。

### 例

REVOKE 文の構文は、GRANT 文と似ています。ユーザー M_Haneef に付与された my_procedure を実行するためのパーミッションを取り消すには、次の文を実行します。

```
REVOKE EXECUTE
ON my_procedure
FROM M_Haneef;
```

sample_table からローを削除するためのパーミッションを取り消すには、次のコマンドを実行します。

```
REVOKE DELETE
ON sample_table
FROM M_Haneef;
```

### 参照


- 「パーミッションの取り消し」533 ページ
- 「REVOKE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』


## パーミッションの取り消し

データベースオブジェクトのユーザーパーミッションを取り消すには、REVOKE 文または Sybase Central のいずれかを使用します。

Sybase Central では、ユーザーごとまたはデータベースオブジェクトごとでパーミッションを表示または変更できます。たとえば、特定のテーブルに対するパーミッションを誰が持っているのかを確認する場合、左側のウィンドウ枠にあるテーブルをクリックし、右側のウィンドウ枠にあるその **[パーミッション]** タブをクリックすることによって、テーブルのパーミッションを表示できます。同様に、特定のユーザーに対するすべてのパーミッションを変更する場合、左側のウィンドウ枠にあるユーザーをクリックすることで、右側のウィンドウ枠でその **[テーブルのパーミッション]** をクリックすることによって、すべてのユーザーの明示的に設定されたテーブルパーミッションを表示できます。

**♦ テーブル、ビュー、カラム、プロシージャー、DB 領域、シーケンスジェネレーターからパーミッションを取り消すには、次の手順に従います (Sybase Central の場合)。**

1. **[SQL Anywhere 12]** プラグインを使用して、DBA 権限を持つユーザーとしてまたはパーミッションを取り消したいデータベースオブジェクトを所有しているユーザーとしてデータベースに接続します。

> **注意**
> 接続パーミッションまたは別のユーザーからのテーブルパーミッションを取り消すとき、他のユーザーはそのデータベースに接続できません。

2. 左側のウィンドウ枠で、ユーザーパーミッションを取り消したいデータベースオブジェクトをクリックするか、パーミッションを取り消したいユーザー (またはグループ) をクリックします。

   - **ユーザーごとにパーミッションを取り消すには**　ユーザー (またはグループ) をクリックした場合、右側のウィンドウ枠で、パーミッションを取り消したいデータベースオブジェクトに対応するパーミッションタブをクリックします。たとえば、テーブルへのパーミッションを取り消すには、**[テーブルのパーミッション]** タブをクリックします。

     パーミッションタブには、ユーザー (またはグループ) が明示的にパーミッションを設定しているデータベースオブジェクト名のリストが表示されています。

   - **データベースオブジェクトごとにパーミッションを取り消すには**　データベースオブジェクトをクリックした場合、右側のウィンドウ枠で **[パーミッション]** タブをクリックします。

     **[パーミッション]** タブでは、データベースオブジェクトに現在パーミッションを持っているユーザーとグループのリストが表示されます。

3. データベースオブジェクトまたは変更しようとしているパーミッションを持つユーザーのローを選択します。

   - **DB 領域、プロシージャー、関数、シーケンスジェネレーター、ビューからパーミッションを取り消すには**　カラムのチェックマークは、ユーザーがパーミッションを持っていることを示します。

     パーミッションを取り消すには、現在チェックマークを含んでいるカラムをクリックして、そのカラムにチェックマークがなくなるまでクリックし続ける必要があります。

   - **テーブルおよびテーブルカラムからパーミッションを取り消すには**　カラムのチェックマークは、ユーザーがパーミッションを持っていることを示します。プラス記号 (+) 付きのチェックマークは、ユーザーがそのパーミッションを持っていることに加えて、そのパーミッションを他のユーザーに付与できることを示しています。プラス記号 (+) 付きのチェックマークは、WITH GRANT OPTION 句付きの GRANT 文を実行することと同等です。

     テーブルを選択するときに、対応する **[カラムのパーミッション]** テーブルが **[パーミッション]** テーブルの下に表示されます。

     **テーブルからパーミッションを取り消すには**　パーミッションを取り消すには、現在チェックマークまたは + (プラス記号) 付きのチェックマークを含んでいるカラムをクリックして、そのカラムにチェックマークがなくなるまでクリックし続ける必要があります。

     テーブルからパーミッションを取り消すときには、テーブルのそのパーミッションに関連するカラムのパーミッションも取り消すことになります。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**テーブルカラムからパーミッションを取り消すには**　パーミッションを取り消すには、現在チェックマークまたは + (プラス記号) 付きのチェックマークを含んでいるカラムをクリックして、そのカラムにチェックマークがなくなるまでクリックし続ける必要があります。

**[カラムのパーミッション]** テーブルからパーミッションを取り消すと、**[付与されている、または付与オプション付きで付与されているカラムパーミッション]** アイコンが前述の **[テーブルのパーミッション]** からなくなります。

**注意**

○あるユーザーに対するあるパーミッションを取り消すとき、そのユーザーがそのパーミッションに対して WITH GRANT OPTION を持っていた場合 (プラス記号 (+) 付きのチェックマーク)、そのユーザーがパーミッションを付与した他のすべてのユーザーも自分たちのパーミッションを取り消されることになります。「WITH GRANT OPTION の取り消し」533 ページを参照してください。

○変更を元に戻すには、ローを選択して **[編集]** » **[元に戻す]** をクリックします。

○パーミッションのテーブルに対して **[凡例]** をオンまたはオフにするには、**[ファイル]** » **[凡例を表示]** をクリックします。

4. すべての変更を保存するには、**[ファイル]** » **[保存]** をクリックします。

**♦ テーブル、ビュー、カラム、プロシージャー、DB 領域、シーケンスジェネレーターからパーミッションを取り消すには、次の手順に従います (SQL の場合)。**

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. REVOKE 文を実行してパーミッションを割り当てます。

**参照**

- 「REVOKE 文」『SQL Anywhere サーバー SQL リファレンス』

## SQL Remote 用のリモートパーミッションの付与および取り消し

**♦ ユーザーに REMOTE パーミッションを付与するには、次の手順に従います (Sybase Central の場合)。**

Sybase Central では、ユーザーとグループの REMOTE パーミッションを管理できます。REMOTE パーミッションを使用すると、通常のユーザーとグループを SQL Remote レプリケーション設定においてリモートユーザーにすることができます。これにより、統合データベースとレプリケーションメッセージを交換できます。

REMOTE パーミッションは、データベースにメッセージタイプを少なくとも 1 つは定義しないかぎり、ユーザーまたはグループに付与することはできません。

REMOTE パーミッションをグループに付与するには、グループ内のすべてのユーザーに REMOTE パーミッションを明示的に与える必要があります。グループのメンバーは、REMOTE パーミッションを継承しません。

ユーザーに REMOTE パーミッションを付与すると、パブリケーションにユーザーのサブスクリプションを作成できます。

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。
2. **[ユーザーとグループ]** をクリックします。
3. ユーザーを右クリックし、**[リモートユーザーに変更]** をクリックします。
4. フィールドに値を入力し、**[OK]** をクリックします。

**♦ リモートユーザーから REMOTE パーミッションを取り消すには、次の手順に従います (Sybase Central の場合)。**

REMOTE パーミッションを取り消すと、リモートユーザーは通常のユーザーに戻ります。REMOTE パーミッションを取り消すと、そのユーザーのサブスクリプションがすべてのパブリケーションから自動的に削除されます。

1. DBA 権限のあるユーザーとしてデータベースに接続します。
2. **[ユーザーとグループ]** または **[SQL Remote ユーザー]** をクリックします。
3. ユーザーを右クリックして、**[リモートの取り消し]** をクリックします。
4. **[はい]** をクリックします。

**参照**

- 「REMOTE パーミッション」『SQL Remote』
- 「SQL Remote システム」『SQL Remote』

# 権限

SQL Anywhere において、「権限」はデータベースレベルのパーミッションととらえることができます。このため、権限はデータベース内の特定のオブジェクト (ユーザー以外) に関連付けられている必要はありません。たとえば、バックアップ権限を持つユーザーはデータベースをバックアップできます。権限にはデータベースオブジェクトパーミッションを含めることもできます。たとえば、プロファイル権限を持つユーザーは、DBA 権限を持つユーザー以外は使用できないシステムテーブルやシステムプロシージャーを使用して、アプリケーションプロファイリングやデータベースのトレーシングタスクを実行できます。

## 権限の継承

次の表に、ユーザーに割り当てることができる権限と、グループメンバーシップを通して継承可能かどうかを示します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 権限 | グループメンバーシップを通して継承可能 | その他の情報 |
|---|---|---|
| BACKUP | なし | 「バックアップ権限」537 ページを参照してください。 |
| DBA | なし | 「DBA 権限」538 ページを参照してください。 |
| PROFILE | Yes | 「プロファイル権限」539 ページを参照してください。 |
| READCLIENTFILE | Yes | 「READCLIENTFILE 権限」540 ページを参照してください。 |
| READFILE | Yes | 「READFILE 権限」540 ページを参照してください。 |
| REMOTE DBA | なし | 「REMOTE DBA 権限」540 ページを参照してください。 |
| RESOURCE | なし | 「リソース権限」541 ページを参照してください。 |
| VALIDATE | なし | 「VALIDATE 権限」541 ページを参照してください。 |
| WRITECLIENTFILE | Yes | 「WRITECLIENTFILE 権限」541 ページを参照してください。 |

## バックアップ権限

バックアップ権限によって、ユーザーはアーカイブバックアップやイメージバックアップを使用してデータベースやトランザクションログをバックアップできます。実行するには、BACKUP 文または dbbackup ユーティリティを使用します。バックアップ権限はグループメンバーシップを通して継承されず、DBA 権限を持つユーザーによってのみ付与できます。

**参照**

- 「BACKUP 文」『SQL Anywhere サーバー SQL リファレンス』
- 「バックアップユーティリティ (dbbackup)」913 ページ

## DBA 権限

データベースを作成するときには、1 つの使用可能なユーザー ID も作成されます。デフォルトでは、最初のユーザー ID は **DBA** であり、パスワードは **sql** となります (パスワードでは大文字と小文字が区別されます)。

ユーザー ID DBA には、データベース内で自動的に DBA 権限が設定されます。DBA 権限を持つユーザーは、データベースに関係するすべての作業を実行できます。DBA 権限を持つユーザーは、データベースオブジェクトの作成とこれらのオブジェクトの所有権の他のユーザー ID への割り当て、テーブル構造の変更、新規ユーザー ID の作成、ユーザーからのパーミッションの取り消し、データベースのバックアップなどを行うことができます。

DBA 権限を持つユーザーは、データベース管理者になります。データベース管理者 (または DBA) は、DBA 権限を持つすべてのユーザーを意味します。

DBA 権限を他のユーザー ID に付与または譲渡することができます。DBA 権限はグループメンバーシップを通して継承されません。

**警告**
データに対する不正アクセスを防止するために、CREATE DATABASE 文の DBA USER 句と DBA PASSWORD 句を使用するか、dbinit -dba オプションを使用して、データベースを展開する前に DBA ユーザーのパスワード (または DBA ユーザーとパスワード) を変更してください。「パスワード」515 ページを参照してください。

**参照**

- 「CREATE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「初期化ユーティリティ (dbinit)」958 ページ
- 「新しいユーザーの作成」513 ページ

### DBA 認証

max_failed_login_attempts ポリシーオプションは、データベースユーザーの失敗ログインの回数をリダイレクトします。DBA ユーザーがデータベースから無期限にロックアウトされないようにするために、ログインポリシーで定義された指定の失敗接続の回数を超過した場合は、アカウントが 1 分間ロックされます。

データベースを再起動して、1 分間のロックアウトを回避することができます。データベースを再起動すると、ログインポリシーで許容されている失敗ログインの最大回数を超えている場合でも、DBA 権限を持つユーザーは 1 回のログインを試行することができます。このログインに失敗すると、ユーザーはデータベースに接続を試みる前に 1 分間待つ必要があります。DBA 権限

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

を持つ別のユーザーが、ALTER USER...RESET LOGIN POLICY 文を使用してアカウントのロックを手動で解除することができます。

ベストプラクティスとして、またアクティブな DBA パスワードが危険にさらされたり、紛失や失念を防ぐために、追加の DBA ユーザーを作成して、そのユーザーのパスワードを安全なロケーションに格納することができます。アクティブな DBA アカウントが危険にさらされた場合は、追加 DBA ユーザーの 1 人としてログインして、危険にさらされた DBA アカウントを削除することができます。

**参照**


- 「ログインポリシー」553 ページ
- 「新しいログインポリシーの作成」557 ページ
- 「既存のユーザーへのログインポリシーの割り当て」558 ページ
- 「ALTER USER 文」『SQL Anywhere サーバー SQL リファレンス』


## プロファイル権限

プロファイル権限によって、ユーザーは次のプロファイリング、トレーシング、診断操作を実行できます。

- アプリケーションプロファイリング
- 診断トレーシング
- プロシージャープロファイリング
- 要求ロギング (要求ログの作成と分析)
- インデックスコンサルタントの実行

プロファイル権限はグループメンバーシップを通して継承できません。

プロファイル権限を持つユーザーは、診断グループに自動的に追加されます。

プロシージャープロファイリングと要求ロギングを実行するには、sa_server_option システムプロシージャーを使用して設定する必要があります。このプロシージャーへのアクセス権限は、プロファイル権限を持つユーザーに付与されていますが、プロシージャープロファイリングと要求ロギングに関連するオプションのみ使用できます。

アプリケーションプロファイリングと診断トレーシングを実行する場合、(DBA 権限ではなく) プロファイル権限を持つユーザーは、データベースのアンロードに必要な特定のパーミッションを持っていないかぎり、別のデータベースを作成してプロファイリングとトレーシングのデータを保存することはできません。ただし、それらのデータを同じデータベースまたはすでに接続している別のデータベースに保存することはできます。

この権限を付与するには、データベースを SQL Anywhere 11 以降のデータベースサーバーで作成しておくか、アップグレードユーティリティ (dbupgrad) または ALTER DATABASE UPGRADE 文を使用してバージョン 11 以降のデータベースにアップグレードしておく必要があります。「アップグレードユーティリティ (dbupgrad)」1061 ページと「ALTER DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

**参照**

- 「アプリケーションプロファイリング」『SQL Anywhere サーバー SQL の使用法』
- 「診断トレーシング」『SQL Anywhere サーバー SQL の使用法』
- 「インデックスコンサルタント」『SQL Anywhere サーバー SQL の使用法』
- 「システムプロシージャーを使用したプロシージャープロファイリング」『SQL Anywhere サーバー SQL の使用法』
- 「要求ロギング」『SQL Anywhere サーバー SQL の使用法』
- 「DBA 権限」538 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』

## READCLIENTFILE 権限

READCLIENTFILE 権限によって、ユーザーはクライアントコンピューターのファイルを読み込む (たとえば、クライアントコンピューターのファイルからデータをロードする) ことができます。

READCLIENTFILE 権限は、グループメンバーシップを通して継承できます。

**参照**

- 「クライアントコンピューター上のデータへのアクセス」『SQL Anywhere サーバー SQL の使用法』
- 「WRITECLIENTFILE 権限」541 ページ
- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「LOAD TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「READ_CLIENT_FILE 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』

## READFILE 権限

READFILE 権限によって、ユーザーは SELECT 文の OPENSTRING 句を使用してファイルを読み込むことができます。READFILE 権限がなくても、ユーザーは OPENSTRING 句を使用して文字列や BLOB 値を問い合わせることはできますが、ファイルの問い合わせはできません。

READFILE 権限は、グループメンバーシップを通して継承できます。

SELECT 文での OPENSTRING 句の使用方法については、「FROM 句」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

## REMOTE DBA 権限

REMOTE DBA 権限は、SQL Remote または Mobile Link の同期ユーザーに限られた DBA パーミッションを付与します。REMOTE DBA 権限は、完全な DBA 権限を付与しなくて済むようにして、DBA のユーザー ID やパスワードの配布に関連するセキュリティ問題を回避します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

REMOTE DBA 権限の使用の詳細については、「GRANT REMOTE DBA 文 [Mobile Link] [SQL Remote]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

## リソース権限

リソース権限によって、ユーザーはテーブル、ビュー、ストアドプロシージャー、トリガーなどのデータベースオブジェクトを作成できます。リソース権限はグループメンバーシップを通して継承されず、DBA 権限を持つユーザーによってのみ付与できます。

トリガーを作成するには、ユーザーにはリソース権限と対象テーブルの ALTER パーミッションの両方が必要です。

別の所有者のデータベースオブジェクトを作成するためには、DBA 権限が必要です。

## VALIDATE 権限

VALIDATE 権限によって、ユーザーはデータベース、テーブル、インデックス、チェックサムの検証を実行できます。実行するには、VALIDATE 文または dbvalid ユーティリティを使用します。VALIDATE 権限はグループメンバーシップを通して継承されず、DBA 権限を持つユーザーによってのみ付与できます。

**参照**

- 「VALIDATE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「検証ユーティリティ (dbvalid)」1064 ページ

## WRITECLIENTFILE 権限

WRITECLIENTFILE 権限によって、ユーザーはクライアントコンピューターにファイルを書き込む (たとえば、UNLOAD TABLE 文を使用してクライアントコンピューターにデータを書き込む) ことができます。

WRITECLIENTFILE 権限は、グループメンバーシップを通して継承できます。

**参照**

- 「クライアントコンピューター上のデータへのアクセス」『SQL Anywhere サーバー SQL の使用法』
- 「READCLIENTFILE 権限」540 ページ
- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「READ_CLIENT_FILE 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』
- 「WRITE_CLIENT_FILE 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』
- 「UNLOAD 文」『SQL Anywhere サーバー SQL リファレンス』

## 権限の付与

**♦ ユーザー ID に権限を付与するには、次の手順に従います (SQL の場合)。**

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. 付与する権限とそれが割り当てられるユーザーの ID を指定して、GRANT 文を実行します。

**参照**

SQL Anywhere でサポートされている権限の詳細については、「権限」536 ページを参照してください。

**例**

たとえば、DBA 権限を付与するための SQL 文は次のようになります。

```
GRANT DBA TO user-id;
```

## 接続されているユーザー

Sybase Central を使用すると、データベースに接続されたすべてのユーザーを追跡することができます。接続されたユーザーのプロパティを表示したり、必要に応じてその接続を切断したりできます。

また、SQL Anywhere モニターの **[接続]** ウィジェットからデータベースに接続しているすべてのユーザーのリストを表示できます。「ウィジェット」1244 ページを参照してください。

**♦ データベースに接続しているすべてのユーザーをリストするには、次の手順に従います (Sybase Central の場合)。**

● 左ウィンドウ枠でデータベースを選択し、右ウィンドウ枠の **[接続]** タブをクリックします。

このタブには、接続に使用したアプリケーション (Sybase Central、Interactive SQL、カスタムクライアントアプリケーションなど) に関係なく、データベースに現在接続しているすべてのユーザーが表示されます。

**♦ ユーザーとデータベースの接続に関するプロパティを検査するには、次の手順に従います (Sybase Central の場合)。**

1. 左ウィンドウ枠でデータベースを選択し、右ウィンドウ枠の **[接続]** タブをクリックします。
2. ユーザーを右クリックし、**[プロパティ]** をクリックします。
3. ユーザーのプロパティを確認し、**[OK]** をクリックします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

# データベースとの接続の切断

### 現在のユーザーとの接続の切断

**♦ データベースの接続を切断するには、次の手順に従います (Sybase Central の場合)。**

1. データベースを選択します。
2. **[ファイル]** » **[切断]** をクリックします。

**♦ データベースの接続を切断するには、次の手順に従います (SQL の場合)。**

● DISCONNECT 文を実行します。

### その他のユーザーとの接続の切断

**♦ ユーザーをデータベースとの接続から切断するには、次の手順に従います (Sybase Central の場合)。**

1. 左ウィンドウ枠でデータベースを選択し、右ウィンドウ枠の **[接続]** タブをクリックします。
2. ユーザーを右クリックし、**[切断]** を選択します。

**♦ 別のユーザーとデータベースの接続を切断するには、次の手順に従います (SQL の場合)。**

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. sa_conn_info システムプロシージャーを使用して、接続するユーザーの接続 ID を決定します。
3. DROP CONNECTION 文を実行します。

**♦ データベースとユーザーの接続を切断するには、次の手順に従います (SQL Anywhere コンソールユーティリティの場合)。**

1. SQL Anywhere コンソールユーティリティからデータベースに接続します。
2. **[ユーザー ID]** カラムでユーザーを右クリックし、**[接続の削除]** をクリックします。

   SQL Anywhere コンソールに表示されるカラムは、**[オプション]** ウィンドウで設定できます。このウィンドウへは **[ファイル]** » **[オプション]** をクリックするとアクセスできます。

**♦ データベースとユーザーの接続を切断するには、次の手順に従います (SQL Anywhere モニターの場合)。**

● 「モニターを使用した接続の切断」1247 ページを参照してください。

### 参照

- 「DISCONNECT 文 [ESQL] [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「DROP CONNECTION 文」『SQL Anywhere サーバー SQL リファレンス』
- 「SQL Anywhere コンソールユーティリティ (dbconsole)」905 ページ
- 「SQL Anywhere コンソールユーティリティ (dbconsole)」1021 ページ

### 例

次の文は、Interactive SQL で DISCONNECT 文を使用して、現在の接続 conn1 を切断する方法を示します。

```
DISCONNECT conn1;
```

次の文は、Embedded SQL 内の DISCONNECT の使用法を示します。

```
EXEC SQL DISCONNECT :conn-name
```

次の文は、接続番号 4 を削除します。

```
DROP CONNECTION 4;
```

# グループ

グループは、メンバーを持つことができるなどの特殊なパーミッションを持つユーザー ID と考えることができます。グループのパーミッションや権限の付与と取り消しは、ユーザーの場合とまったく同じように行います。

グループの階層を構成し、各グループを別のグループのメンバーにできます。ユーザー ID は複数のグループに属することができます。ユーザーとグループの関係は多対多です。

パスワードなしのグループも作成できます。これによって、グループのユーザー ID を使用したデータベースへの接続を防ぐことができます。「パスワードのないグループ」550 ページを参照してください。

グループの権限とパーミッションを管理するには、ユーザーのパーミッションと権限を管理する場合と同じ手順に従います。「ユーザー」512 ページを参照してください。

グループの REMOTE パーミッションの管理については、「SQL Remote 用のリモートパーミッションの付与および取り消し」535 ページを参照してください。

### パーミッションと権限の継承

ユーザーとグループは、メンバーであるグループの継承可能なすべてのパーミッションを継承します。同じパーミッションを他のユーザーに付与するパーミッション (GRANT....WITH GRANT OPTION 文を使用) は、グループのメンバーには継承されません。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

グループのメンバーは、属しているグループの次のパーミッションだけを継承できます。

- ALL
- ALTER
- DELETE
- INSERT
- REFERENCES
- SELECT
- UPDATE

グループのメンバーは、属しているグループに設定されている次の権限だけを継承できます。

- READCLIENTFILE
- READFILE
- WRITECLIENTFILE

**例**

次の例では、group1 と group2 の 2 つのグループを作成します。ユーザー bobsmith を両方のグループに作成し、メンバーシップを付与します。テーブル table1 を作成し、この新しいテーブルに対する SELECT パーミッションと INSERT パーミッションを group2 に付与します。

```
GRANT CONNECT, GROUP TO group1;
GRANT CONNECT, GROUP TO group2;
GRANT CONNECT TO bobsmith IDENTIFIED BY sql;
GRANT MEMBERSHIP IN GROUP group1 TO bobsmith;
GRANT MEMBERSHIP IN GROUP group2 TO bobsmith;

CREATE TABLE DBA.table1( column1 INT, modified_by VARCHAR(128) DEFAULT USER );
GRANT SELECT, INSERT ON DBA.table1 TO group2;
```

bobsmith は group2 のメンバーなので、table1 に対する SELECT パーミッションと INSERT パーミッションを継承し、次に示すようにテーブルに値を挿入できます。

```
CONNECT USER bobsmith IDENTIFIED BY sql;
INSERT INTO DBA.table1(column1) VALUES(1);
```

## ユーザーパーミッションの評価方法

グループによって個々のユーザーのパーミッションは複雑になります。たとえば、ユーザー M_Haneef が、あるテーブルに対して SELECT と UPDATE のパーミッションを個人で所有しており、2 つのグループのメンバーでもあるとします。一方のグループではテーブルに対するパーミッションをまったく持たず、もう一方のグループでは SELECT パーミッションだけを持つとします。この場合、このユーザーのパーミッションは一体どうなるでしょうか。

SQL Anywhere は、ユーザー ID が特定のアクションを実行するパーミッションを持っているかどうかを、次の順で判断します。

1. ユーザー ID が DBA 権限を持っている場合、そのユーザーはデータベース内ですべての作業を実行できます。

2. DBA 権限がない場合、パーミッションは個別のユーザーに与えられたパーミッションによって異なります。ユーザー ID にその作業を実行するパーミッションが付与されていれば、その作業は実行できます。

3. ユーザーに対して個別の設定が行われていない場合、パーミッションはそのユーザーが属する各グループのパーミッションによって異なります。いずれかのグループがその作業を実行するパーミッションを持っていれば、そのユーザーもメンバーとしてパーミッションを持っていることになり、その作業を実行できます。

このようにして、パーミッションが設定されている順序に関する問題を最小限に抑えています。

## グループの作成

**♦ 新しいプロシージャーを作成するには、次の手順に従います (Sybase Central の場合)。**

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。

2. **[ユーザーとグループ]** を右クリックし、**[新規]** » **[グループ]** をクリックします。

3. **グループ作成ウィザード**の指示に従います。

4. ユーザーをグループに追加します。 「グループメンバーシップを既存のユーザーまたはグループに付与する」547 ページを参照してください。

**♦ 新しいグループを作成するには、次の手順に従います (SQL の場合)。**

1. DBA 権限のあるユーザーとして、データベースに接続します。

2. GRANT GROUP TO 文を実行します。この文に指定したユーザー ID がまだ作成されていない場合、文は失敗します。

3. メンバーをグループに追加します。「グループメンバーシップを既存のユーザーまたはグループに付与する」547 ページを参照してください。

**参照**

- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「CREATE USER 文」『SQL Anywhere サーバー SQL リファレンス』
- 「新しいユーザーの作成」513 ページ

**例**

ユーザー ID personnel を作成します。

```
CREATE USER personnel
IDENTIFIED BY group_password;
```

ユーザー ID personnel をグループにします。

```
GRANT GROUP TO personnel;
```

## グループメンバーシップを既存のユーザーまたはグループに付与する

**♦ ユーザーまたはグループを別のグループに追加するには、次の手順に従います (Sybase Central の場合)。**

グループ内でメンバーシップを与えられたユーザーは、そのグループに関連した継承可能なパーミッションと権限を継承します。「パーミッションと権限の継承」544 ページを参照してください。

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。
2. **[ユーザーとグループ]** をクリックします。
3. ユーザーまたはグループをダブルクリックします。
4. ユーザーの場合
   - **[ファイル]** » **[新規]** » **[メンバーシップ]** をクリックします。

   グループの場合
   - **[メンバーシップ]** タブをクリックします。
   - **[ファイル]** » **[新規]** » **[メンバーシップ]** をクリックします。
5. **[名前]** リストでグループをダブルクリックします。

   右ウィンドウ枠の **[メンバーシップ]** タブに新しいグループが表示されます。

**♦ ユーザーまたはグループを別のグループに追加するには、次の手順に従います (SQL の場合)。**

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. グループとユーザーを指定した GRANT MEMBERSHIP IN GROUP 文を実行します。

### 参照

- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「新しいユーザーの作成」513 ページ

### 例

ユーザー M_Haneef に、グループ personnel のメンバーシップを付与します。

```
GRANT MEMBERSHIP
IN GROUP personnel
TO M_Haneef;
```

# グループメンバーシップの取り消し

**♦ ユーザーまたはグループを別のグループから取り除くには、次の手順に従います (Sybase Central の場合)。**

グループからユーザーまたはグループを取り除いても、データベース (または他のグループ) からは削除されません。データベースから削除するには、ユーザーまたはグループ自体を削除する必要があります。

グループにユーザーを追加すると、ユーザーはそのグループに割り当てられたすべてのパーミッションを継承します。SQL Anywhere では、ユーザーがグループのメンバーとして継承するパーミッションのサブセットを取り消すことはできません。取り消すことができるのは、GRANT 文によって明示的に付与されるパーミッションだけです。異なるユーザーに別々のパーミッションを付与する必要がある場合、適切なパーミッションを持つグループを別々に作成するか、必要なパーミッションを各ユーザーに明示的に付与できます。

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。
2. **[ユーザーとグループ]** をクリックします。
3. ユーザーまたはグループをダブルクリックします。
4. **[メンバーシップ]** ウィンドウ枠でグループを右クリックし、**[メンバーシップの削除]** をクリックします。

**ヒント**
グループをダブルクリックし、右ウィンドウ枠の **[メンバー]** タブをクリックし、ユーザーまたはグループを右クリックし、**[メンバーの削除]** をクリックしてユーザーを取り除くこともできます。

**♦ ユーザーまたはグループを別のグループから取り除くには、次の手順に従います (SQL の場合)。**

グループからユーザーまたはグループを取り除いても、データベース (または他のグループ) からは削除されません。データベースから削除するには、ユーザーまたはグループ自体を削除する必要があります。

グループにユーザーを追加すると、ユーザーはそのグループに割り当てられたすべてのパーミッションを継承します。SQL Anywhere では、ユーザーがグループのメンバーとして継承するパーミッションのサブセットを取り消すことはできません。取り消すことができるのは、GRANT 文によって明示的に付与されるパーミッションだけです。異なるユーザーに別々のパーミッションを付与する必要がある場合、適切なパーミッションを持つグループを別々に作成するか、必要なパーミッションを各ユーザーに明示的に付与することができます。

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. グループとユーザー名を指定した REVOKE MEMBERSHIP IN GROUP 文を実行します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照

- 「REVOKE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「新しいユーザーの作成」513 ページ
- 「ユーザーをデータベースから削除する」517 ページ
- 「グループをデータベースから削除する」552 ページ

### 例

グループ personnel からユーザー M_Haneef を取り除きます。

```
REVOKE MEMBERSHIP
IN GROUP personnel
FROM M_Haneef;
```

## グループのパーミッションと権限

グループへのパーミッションは、通常のユーザー ID とまったく同じ方法で付与します。テーブル、ビュー、プロシージャーに対するパーミッションは、他のグループとそのメンバーも含めて、グループのメンバーに継承されます。

データベースオブジェクトの所有者は単一のユーザー ID に属し、グループメンバーには受け継がれません。ユーザー ID personnel がテーブルを作成した場合は、ユーザー ID personnel がそのテーブルの所有者となり、テーブルに変更を加える権限や、テーブルに関する権限を他のユーザーに与える権限を保持します。personnel のメンバーである他のユーザー ID は、このテーブルの所有者ではなく、これらの権利を持ちません。付与されたパーミッションのみを継承します。たとえば、DBA 権限を持つユーザーまたはユーザー ID personnel が、テーブルに対する SELECT パーミッションをユーザー ID personnel に明示的に付与した場合は、すべてのグループメンバーが、そのテーブルに対する SELECT アクセス権を継承します。

グループには一部の権限も付与できます。

**注意**

- WITH GRANT OPTION はユーザーにのみ指定できます。グループのメンバーは、グループに付与されている WITH GRANT OPTION を継承しません。
- グループのメンバーは、DBA、RESOURCE、GROUP の各権限を継承しません。ユーザー ID が リソース権限を持っている場合でも、personnel のメンバーにはリソース権限がありません。

### 参照

- 「テーブルパーミッション」526 ページ

## グループが所有するテーブルの参照

データベース中のテーブルとプロシージャーを検索するのにグループを使うことができます。すべてのユーザーがグループ PUBLIC に属し、グループ PUBLIC が SYSGROUPS ビューを所有

する SYS グループに属するため、次に示すクエリは常にビュー SYS.SYSGROUPS を検索します。

```
SELECT * FROM SYSGROUPS;
```

SYSGROUPS ビューには、データベース内のグループメンバーシップを示す *group-name* と *member-name* のペアのリストが含まれています。

テーブル employees がユーザー ID personnel によって所有されており、**M_Haneef** が personnel グループのメンバーである場合、**M_Haneef** は SQL 文で単に employees と指定することでテーブル employees を参照できます。personnel グループのメンバーでないユーザーは、「修飾された」名前 personnel.employees を使用する必要があります。

### テーブルを所有するグループの作成

名前を修飾しないでテーブルにアクセスできるように、テーブルを所有することだけが目的のグループを作成することをおすすめします。このグループには何のパーミッションも与えませんが、すべてのユーザーをこのグループのメンバーにします。次にパーミッショングループを作成し、ユーザーを適宜それらのパーミッショングループのメンバーにします。

ユーザーがグループ所有のテーブルと同じ名前のテーブルを所有している場合、修飾されていないテーブル名は、グループが所有するテーブルではなくユーザーが所有するテーブルを表します。同様に、ユーザーが同じ名前のテーブルを所有する複数のグループに属している場合、そのユーザーはテーブル名を修飾する必要があります。

「データベースオブジェクトの名前とプレフィクス」562 ページを参照してください。

## パスワードのないグループ

グループのユーザー ID に属すユーザーには、何らかのパーミッションがあります。グループに属しているユーザーは、グループのユーザー ID で作成されたデータベース内のすべてのテーブルに対して、所有者のパーミッションを持ちます。

他のユーザー ID がグループメンバーシップを変更できるようにするのではなく、DBA だけがグループとグループのデータベースオブジェクトを処理するようにデータベースを設定できます。この場合は、グループの作成時にそのグループのユーザー ID で接続できないよう指定します。そのためには、次のように CREATE USER 文をパスワードなしで入力します。次の文は、ユーザー ID personnel を作成します。

```
CREATE USER personnel;
```

このユーザー ID にグループパーミッションを付与し、他のユーザー ID にグループのメンバーシップを付与して personnel に与えられているパーミッションを継承させることができます。ただし、ユーザー ID personnel には有効なパスワードがないので、このユーザー ID を使用してデータベースに接続することはできません。

ユーザーがユーザー ID personnel を使用してデータベースに接続できない場合でも、このユーザー ID はデータベースオブジェクトの所有者になることができます。CREATE TABLE 文、CREATE PROCEDURE 文、CREATE VIEW 文では、文を実行する以外のユーザーとしてオブジェ

クトの所有者を指定できます。このような所有者の割り当てを実行できるのは、DBA 権限を持つユーザーだけです。

## 特殊なグループとユーザー

データベースを作成すると、SYS、PUBLIC、dbo の各グループも自動的に作成されます。どのグループもパスワードを持たないため、SYS、PUBLIC、dbo でデータベースに接続することはできません。しかし、これらのグループはデータベース中で重要な働きをします。「システムビューに対するユーザーとパーミッション」518 ページを参照してください。

- **SYS グループ**　SYS グループは、全データベースオブジェクトと全ユーザー ID を含む、完全なデータベーススキーマを記述するシステムテーブルとビューを所有します。

  システムテーブルとビューの詳細およびテーブルへのアクセス権については、「テーブル」『SQL Anywhere サーバー SQL リファレンス』と「システムビュー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

- **PUBLIC グループ**　PUBLIC グループは、システムテーブルに対する SELECT パーミッションを持ちます。また、PUBLIC グループは SYS グループのメンバーであり、システムテーブルとビューへの読み込みアクセス権を受け継いでいます。したがって、データベースのユーザーは誰でもデータベーススキーマについて知ることができます。このアクセスを制限するには、PUBLIC から SYS グループのメンバーシップを取り消します。

  新しいユーザー ID は自動的に PUBLIC グループのメンバーとなり、DBA 権限を持つユーザーによってそのグループ明示的に付与されたパーミッションをすべて継承します。必要に応じてユーザーごとに PUBLIC グループのメンバーシップを取り消すこともできます。

- **DIAGNOSTICS グループ**　DIAGNOSTICS グループは、診断テーブルやビューを所有しています。このグループのメンバーは、パーミッションを継承して診断操作を実行します。「パフォーマンスの向上、診断、モニタリング」『SQL Anywhere サーバー SQL の使用法』を参照してください。

- **dbo グループ**　dbo グループは、多数のシステムストアドプロシージャーやビューを所有しています。dbo グループは、SYS グループのメンバーです。PUBLIC グループは、dbo グループのメンバーです。dbo グループは、Ultra Light と Mobile Link に使用するテーブルも所有しています。

- **SA_DEBUG グループ**　SA_DEBUG グループのメンバーシップが与えられたユーザーは、**SQL Anywhere デバッガー**を使用できます。「SQL Anywhere のデバッガー」『SQL Anywhere サーバー SQL の使用法』を参照してください。

- **SYS_SPATIAL_ADMIN_ROLE グループ**　このグループのメンバーシップが与えられたユーザーは、空間参照系と空間計測単位を作成、変更、または削除できます。「空間データの概要」『SQL Anywhere サーバー 空間データサポート』を参照してください。

- **EXTENV_MAIN ユーザーと EXTENV_WORKER ユーザー**　EXTENV_MAIN と EXTENV_WORKER は、外部環境のサポートのために使用されます。これらのユーザーは内部でのみ使用され、パーミッションは制限されます。「SQL Anywhere 外部環境のサポート」『SQL Anywhere サーバー プログラミング』を参照してください。

# グループをデータベースから削除する

**♦ データベースからグループを削除するには、次の手順に従います (Sybase Central の場合)。**

データベースからのユーザーまたはグループの削除は、それらを別のグループから取り除くことと異なります。データベースからグループを削除しても、データベースからそのグループのメンバーは削除されません。ただし、削除されたグループに対するメンバーシップはなくなります。

グループを削除できるのは、DBA 権限を持つユーザーだけです。

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。
2. **[ユーザーとグループ]** をクリックします。
3. グループを右クリックし、**[削除]** をクリックします。
4. **[はい]** をクリックします。

**♦ データベースからグループを削除するには、次の手順に従います (SQL の場合)。**

データベースからのユーザーまたはグループの削除は、それらを別のグループから取り除くことと異なります。データベースからグループを削除しても、データベースからそのグループのメンバーは削除されません。ただし、削除されたグループに対するメンバーシップはなくなります。

グループを削除できるのは、DBA 権限を持つユーザーだけです。

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. REVOKE CONNECT FROM 文を実行します。

**参照**

- 「REVOKE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「ユーザーパーミッションの取り消し」532 ページ
- 「ユーザーをデータベースから削除する」517 ページ

**例**

データベースからグループ personnel を削除します。

```
REVOKE CONNECT FROM personnel;
```

# 接続リソースの使用

ユーザーとグループのセットを作成すると、データベースのパーミッションを管理できます。またデータベースのセキュリティと管理によって、個々のユーザーが使用できるリソースを制限することもできます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

たとえば、他のユーザーのデータベースへの接続速度を落とさないように、1 つの接続がメモリや CPU を大量に使用することを防止できます。

SQL Anywhere には、DBA 権限を持つユーザーがリソースの制御に使用できるデータベースオプションが備わっています。このオプションを「リソースガバナー」といいます。

### オプションの設定

SET OPTION 文を次のように使用して、データベースオプションを設定します。

**SET** [ **TEMPORARY** ] **OPTION** ... [ *userid*. | **PUBLIC**. ]*option-name* = [ *option-value* ]

### 管理できるリソース

次のオプションを使用して、リソースを管理できます。

- **max_cursor_count**　接続用のカーソルの数を制限します。
- **max_statement_count**　接続用の準備文の数を制限します。
- **priority**　接続からの要求を実行する優先度レベルを設定します。
- **max_priority**　接続の最高優先度レベルを制御します。

データベースオプション設定は、グループ構造に継承されません。

### 参照


- 「max_cursor_count オプション」642 ページ
- 「max_statement_count オプション」646 ページ
- 「priority オプション」664 ページ
- 「max_priority オプション」643 ページ
- 「データベースオプション」575 ページ
- 「SET OPTION 文」『SQL Anywhere サーバー SQL リファレンス』


## ログインポリシー

「ログインポリシー」は、ユーザーのデータベース接続を作成するときに適用されるルールセットで構成されます。すべての新しいデータベースには、ルートログインポリシーが含まれています。ルートログインポリシーの値を変更することはできますが、ポリシーは削除できません。ログインポリシーは、ユーザーログインのルールのみを管理し、権限およびパーミッションとは切り離されています。ログインポリシーは、グループメンバーシップを通して継承されません。

次の表は、ログインポリシーによって管理される設定とルートログインポリシーのデフォルト設定を示します。

| policy-option-name | 説明 | デフォルト値 | 適用対象 |
|---|---|---|---|
| password_life_time | パスワードの変更が必要となるまでの最大日数。 | Unlimited | すべてのユーザー (DBA 権限を持つユーザーを含む)。 |
| password_grace_time | パスワードの有効期限が切れるまでの日数。この期間中は、ログインは許可されますが、デフォルトの post_login プロシージャーから警告が出ます。 | 0 | すべてのユーザー (DBA 権限を持つユーザーを含む)。 |
| password_expiry_on_next_login | このオプションの値が ON の場合、次回のログイン後にユーザーのパスワードは期限切れになります。 | OFF | すべてのユーザー (DBA 権限を持つユーザーを含む)。 |
| locked | このオプションの値が ON の場合、ユーザーは新しい接続を確立できません。DBA 権限を持つユーザーをロックすることはできません。sa_get_user_status システムプロシージャーの reason_locked カラムは、データベースサーバーで生成されたユーザーのロック理由を示す文字列を返します。 | OFF | DBA 権限を持たないユーザー。 |
| max_connections | ユーザーに許容される同時接続の最大数。 | Unlimited | DBA 権限を持たないユーザー。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| policy-option-name | 説明 | デフォルト値 | 適用対象 |
|---|---|---|---|
| max_failed_login_attempts | 前回のログイン成功以降、ユーザーがロックされるまでのログイン失敗の最大回数。 | Unlimited | すべてのユーザー (DBA 権限を持つユーザーを含む)。<br><br>最後のログイン失敗から 1 分が経過すると、DBA ユーザーアカウントのロックは解除されます。 |
| max_days_since_login | 同一ユーザーによる 2 回の連続するログインの間で許容される最大経過日数。 | Unlimited | DBA 権限を持たないユーザー。 |
| max_non_dba_connections | DBA 権限を持たないユーザーが確立できる同時接続の最大数。このオプションは、ルートログインポリシーでのみサポートされます。 | Unlimited | DBA 権限を持たないユーザー。ルートログインポリシーのみ。 |

次の場合、ユーザーにルートログインポリシーが割り当てられます。

- 新しいユーザーを作成し、ログインポリシーを指定していない場合
- アンロードユーティリティ (dbunload) を使用して、以前のバージョンの SQL Anywhere で作成されたデータベースを再構築する場合
- アップグレードユーティリティ (dbupgrad) または ALTER DATABASE UPGRADE 文を使用して、SQL Anywhere バージョン 10 のデータベースをアップグレードする場合

ログインポリシーは、作成、変更、削除できます。また、ユーザーを作成、変更、削除したり、ログインポリシーをユーザーに割り当てたりすることができます。sa_get_user_status システムプロシージャーを使用すると、ユーザーの現在のステータスに関する情報を取得できます。

### ログインポリシー設定の継承

「root」という名前のデフォルトのログインポリシーはデータベースに保存され、すべてのポリシーのデフォルトのオプション値が含まれています。デフォルト以外の設定を使用する場合は、ルートポリシーを変更するか、新しいポリシーを作成してデフォルトの設定を上書きする内容に変更します。ポリシーは、デフォルト設定を変更して上書きしないかぎり、ルートポリシーのデフォルト設定を継承します。

たとえば、ルートポリシーの max_connections の値が 5 であるとします。この場合、myPolicy という名前のポリシーを作成して、max_connections の値を Unlimited に設定します。次に、ユーザーを作成し、そのユーザーに myPolicy ログインポリシーを割り当てます。ユーザーがログイ

ンすると、ログインポリシーオプションの設定はルートログインポリシーから継承されますが、max_connections は Unlimited に設定されたままになります。

ルートポリシーから継承されるデフォルト値について理解しておくことは重要です。これは、ルートポリシーのオプション設定を後で変更すると、デフォルトの設定値に依存しているユーザーのポリシーに影響が出るためです。ルートポリシーの値を変更しても、その設定が上書きされたポリシーを割り当てられているユーザーには影響しません。

**参照**

- 「CREATE LOGIN POLICY 文」『SQL Anywhere サーバー SQL リファレンス』
- 「ALTER LOGIN POLICY 文」『SQL Anywhere サーバー SQL リファレンス』
- 「DROP LOGIN POLICY 文」『SQL Anywhere サーバー SQL リファレンス』
- 「sa_get_user_status システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「sa_get_user_status システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』

## ルートログインポリシーの変更

**♦ ルートログインポリシーを変更するには、次の手順に従います (Sybase Central の場合)。**

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。
2. 左ウィンドウ枠で、**[ログインポリシー]** をクリックします。
3. 右ウィンドウ枠で **[root]** を右クリックし、**[プロパティ]** をクリックします。
4. ポリシー値を変更し、**[OK]** をクリックします。

**♦ ルートログインポリシーを変更するには、次の手順に従います (SQL の場合)。**

1. DBA 権限のあるユーザーとしてデータベースに接続します。
2. ALTER LOGIN POLICY 文を実行します。

**参照**

- 「ALTER LOGIN POLICY 文」『SQL Anywhere サーバー SQL リファレンス』

**例**

次の例では、locked 値と max_connections 値を上書きする設定をルートログインポリシーに作成します。

```
ALTER LOGIN POLICY root
locked=on
max_connections=5;
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## 新しいログインポリシーの作成

**♦ ログインポリシーを作成するには、次の手順に従います (Sybase Central の場合)。**

作成するログインポリシーをユーザーに割り当てないと、ユーザーにはルートログインポリシーが割り当てられます。

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。
2. **[ログインポリシー]** を右クリックし、**[新規]** » **[ログインポリシー]** をクリックします。
3. **ログインポリシー作成ウィザード**の指示に従います。

**♦ ログインポリシーを作成するには、次の手順に従います (SQL の場合)。**

作成するログインポリシーをユーザーに割り当てないと、ユーザーにはルートログインポリシーが割り当てられます。

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. CREATE LOGIN POLICY 文を実行します。すでに存在するログインポリシーを指定すると、文の実行は失敗します。

**参照**

- 「CREATE LOGIN POLICY 文」『SQL Anywhere サーバー SQL リファレンス』
- 「既存のユーザーへのログインポリシーの割り当て」558 ページ
- 「ログインポリシーの変更」559 ページ

**例**

次の例は、オプション値を指定した Test1 ログインポリシーを作成します。

```
CREATE LOGIN POLICY Test1;
```

## ユーザーの作成とログインポリシーの割り当て

**♦ 新しいユーザーを作成してログインポリシーを割り当てるには、次の手順に従います (Sybase Central の場合)。**

ユーザーアカウントを作成してもログインポリシーを割り当てないと、ルートログインポリシーが割り当てられます。

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。
2. **[ユーザーとグループ]** を右クリックし、**[新規]** » **[ユーザー]** をクリックします。

3. **ユーザー作成ウィザード**の指示に従います。

**♦ 新しいユーザーを作成してログインポリシーを割り当てるには、次の手順に従います (SQL の場合)。**

ユーザーアカウントを作成してもログインポリシーを割り当てないと、ルートログインポリシーが割り当てられます。

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. CREATE USER 文を実行します。

**参照**

- 「CREATE USER 文」『SQL Anywhere サーバー SQL リファレンス』
- 「新しいユーザーの作成」513 ページ

**例**

次の例は、パスワードが "welcome" の SQLTester というユーザーを作成し、Test1 ログインポリシーを割り当てます。

```
CREATE USER SQLTester IDENTIFIED BY welcome
LOGIN POLICY Test1;
```

## 既存のユーザーへのログインポリシーの割り当て

**♦ ログインポリシーを既存のユーザーに割り当てるには、次の手順に従います (Sybase Central の場合)。**

カスタマイズしたログインポリシーを割り当てないと、ユーザーにはルートログインポリシーが割り当てられます。次の手順に従って、ユーザーのログインポリシー割り当てを変更します。

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。
2. 左ウィンドウ枠で、**[ユーザーとグループ]** をクリックします。
3. 右ウィンドウ枠でユーザーを右クリックし、**[プロパティ]** をクリックします。
4. **[ログインポリシー]** リストで、ログインポリシーを選択します。
5. **[OK]** をクリックします。

**♦ ログインポリシーを既存のユーザーに割り当てるには、次の手順に従います (SQL の場合)。**

カスタマイズしたログインポリシーを割り当てないと、ユーザーにはルートログインポリシーが割り当てられます。次の手順に従って、ユーザーのログインポリシー割り当てを変更します。

1. DBA 権限のあるユーザーとしてデータベースに接続します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

2. ALTER USER 文を実行します。

**参照**

- 「ALTER USER 文」『SQL Anywhere サーバー SQL リファレンス』
- 「新しいユーザーの作成」513 ページ

**例**

次の例は、Test2 ログインポリシーを SQLTester に割り当てます。

```
ALTER USER SQLTester
LOGIN POLICY Test2;
```

## ログインポリシーの変更

**♦ ログインポリシーを変更するには、次の手順に従います (Sybase Central の場合)。**

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。
2. 左ウィンドウ枠で、**[ログインポリシー]** をクリックします。
3. 右ウィンドウ枠でログインポリシーを右クリックし、**[プロパティ]** をクリックします。
4. ログインポリシーの値を変更します。
5. **[OK]** をクリックします。

**♦ ログインポリシーを変更するには、次の手順に従います (SQL の場合)。**

1. DBA 権限のあるユーザーとしてデータベースに接続します。
2. ALTER LOGIN POLICY 文を実行します。

**参照**

- 「ALTER LOGIN POLICY 文」『SQL Anywhere サーバー SQL リファレンス』
- 「既存のユーザーへのログインポリシーの割り当て」558 ページ

**例**

次の例では、locked 値と max_connections 値を上書きする設定を Test1 ログインポリシーに作成します。

```
ALTER LOGIN POLICY Test1
locked=on
max_connections=5;
```

## ログインポリシーの削除

♦ **ログインポリシーを削除するには、次の手順に従います (Sybase Central の場合)。**

> **注意**
> ユーザーに割り当てられたままのログインポリシーは削除できません。
>
> ユーザーを別のログインポリシーに割り当ててから、カスタマイズしたログインポリシーを削除してください。

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。
2. 左ウィンドウ枠で、**[ログインポリシー]** をクリックします。
3. **[ログインポリシー]** ウィンドウ枠でログインポリシーを右クリックし、**[削除]** をクリックします。
4. **[はい]** をクリックします。

♦ **ログインポリシーを削除するには、次の手順に従います (SQL の場合)。**

> **注意**
> ユーザーに割り当てられたままのログインポリシーは削除できません。
>
> ユーザーを別のログインポリシーに割り当ててから、カスタマイズしたログインポリシーを削除してください。

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. DROP LOGIN POLICY 文を実行します。

**参照**

- 「DROP LOGIN POLICY 文」『SQL Anywhere サーバー SQL リファレンス』

**例**

次の例は、Test1 ログインポリシーを削除します。

```
DROP LOGIN POLICY Test1;
```

## 読み込み専用データベースでのログインポリシーの管理

データベースを読み込み専用モードで起動すると、既存するそのデータベースの永続的な状態に基づいてログインポリシーが使用されます。割り当てるログインポリシーは、現在のセッションにのみ有効です。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

後で読み込み専用モードで起動するデータベースに対してログイン管理が有効な場合、次の制限が適用されます。

●サーバーによるログイン管理は、起動前のデータベース状態に基づいて行われます。

●データベースの状態を明示的に変更する文は拒否され、結果はエラーになります。

●サーバーは、failed_login_attempts や last_login_time などの各ユーザーに関する動的情報を引き続き維持します。ただし、このような情報は一時的なメモリに格納されるため、データベースを停止すると失われます。データベースは、起動前と同じ状態に戻ります。

●既存のログイン管理ポリシーによってアカウントがロックされると、ユーザーはログインできません。また、ログイン時にパスワードを変更する通常の方法は使用できなくなります。

●高可用性システムでミラーデータベースとして動作しているためにデータベースが読み込み専用になっている場合、プライマリデータベースで実行された文の結果はミラーデータベースに反映されます。また、プライマリサーバーで収集された動的情報はミラーデータベースに送信され、ミラーデータベースのために収集された情報とともに一時メモリにマージされます。

●高可用性システムのミラーデータベースまたは読み込み専用のスケールアウトシステムのコピーノードとして動作しているためにデータベースが読み込み専用になっている場合、プライマリデータベースで実行された文の結果は読み込み専用データベースに反映されます。また、プライマリサーバーで収集された動的情報は読み込み専用データベースに送信され、読み込み専用データベースのために収集された情報とともに一時メモリにマージされます。

## ファイル管理文の実行に必要なパーミッション

データベースサーバーのオプション -gu は、誰がファイル管理文を実行できるかを制御します。このオプションを使用して、特定の管理タスクを実行できるユーザーを指定します。「-gu dbeng12/dbsrv12 サーバーオプション」247 ページを参照してください。

ファイル管理文を使用するためのパーミッションには、4 つのレベルがあります。

| -gu オプション | 影響 | 適用対象 |
|---|---|---|
| **all** | 誰でもファイル管理文を実行できる | ユーティリティデータベースを含むすべてのデータベース |
| **none** | 誰もファイル管理文を実行できない | ユーティリティデータベースを含むすべてのデータベース |
| **DBA** | DBA 権限を持つユーザーだけがファイル管理文を実行できる | ユーティリティデータベースを含むすべてのデータベース |

| -gu オプション | 影響 | 適用対象 |
|---|---|---|
| **utility_db** | ユーティリティデータベースに接続できるユーザーだけがファイル管理文を実行できる | ユーティリティデータベースのみ |

**例**

ファイル管理文の使用を禁止するには、-gu オプションの none パーミッションレベルを使用してデータベースサーバーを起動します。次のコマンドを入力すると、データベースサーバーが起動され、サーバー名が TestSrv になります。このコマンドによって *mytestdb.db* データベースがロードされますが、どのユーザーも、そのサーバーを使用してデータベースを作成または削除したり、他のファイル管理文を実行したりすることはできません。これは、ユーザーのリソース作成権の有無や、ユーティリティデータベースをロードして接続できるかどうかには関係ありません。

```
dbsrv12 -n TestSrv -gu none c:¥mytestdb.db
```

ユーティリティデータベースのパスワードを知っているユーザーだけにファイル管理文の実行を許可するには、次のコマンドを実行してサーバーを起動します。「-su dbeng12/dbsrv12 サーバーオプション」276 ページを参照してください。

```
dbsrv12 -n TestSrv -su secret -gu utility_db
```

次のコマンドを使用することによって、Interactive SQL をクライアントアプリケーションとして起動し、サーバー TestSrv に接続して、ユーティリティデータベースをロードし、ユーザーを接続します。

```
dbisql -c "UID=DBA;PWD=secret;DBN=utility_db;Host=host1;Server=TestSrv"
```

上記のコマンドを正常に実行できた場合、ユーザーはユーティリティデータベースに接続し、ファイル管理文を実行できるようになります。

# データベースオブジェクトの名前とプレフィクス

データベースオブジェクトの名前は識別子である必要があります。

有効な識別子の規則については、「識別子」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

このマニュアルに示すクエリと SQL 文のサンプルでは、サンプルデータベース内のデータベースオブジェクトを一般に簡略名で参照しています。次に例を示します。

```
SELECT *
FROM Employees;
```

テーブル、プロシージャー、ビューにはすべて所有者があります。ユーザー ID DBA は、サンプルデータベース内のテーブルを所有します。場合によっては、オブジェクト名に所有者のユーザー ID をプレフィクスとして付ける必要があります。次に例を示します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
SELECT *
FROM DBA.Employees;
```

この Employees テーブルの参照を「修飾された」状態であるといいます。単にオブジェクト名を示すだけでよい場合もあります。この項では、どのような場合にテーブル、ビュー、プロシージャーに所有者名をプレフィクスとして付ける必要があるかを説明します。

データベースオブジェクトを参照するときプレフィクスが必要ないのは、次の場合です。

●自分がデータベースオブジェクトの所有者である場合。

●自分がデータベースオブジェクトを所有するグループのメンバーである場合。

**例**

企業のデータベースを例に説明します。ユーザー ID company がすべてのテーブルを作成し、このユーザー ID はデータベース管理者に属するので、DBA 権限を持ちます。

```
CREATE USER Company
IDENTIFIED BY secret;
GRANT DBA TO Company;
```

ユーザー ID company が、データベース内のテーブルを次のように作成しました。

```
CONNECT USER company IDENTIFIED BY secret;
CREATE TABLE company.Customers ( ... );
CREATE TABLE company.Products ( ... );
CREATE TABLE company.Orders ( ... );
CREATE TABLE company.Invoices ( ... );
CREATE TABLE company.Employees ( ... );
CREATE TABLE company.Salaries ( ... );
```

会社の全員がすべての情報にアクセスできるようにはしません。営業部の 2 人のユーザー ID Joe と Sally に、テーブル Customers、Products、Orders へのアクセス権限を与える場合を考えてみます。これを行うには、Sales グループを作成します。

```
CREATE USER Sally IDENTIFIED BY xxxxx;
CREATE USER Joe IDENTIFIED BY xxxxx;
CREATE USER Sales IDENTIFIED BY xxxxx;
GRANT GROUP TO Sales;
GRANT ALL ON Customers TO Sales;
GRANT ALL ON Orders TO Sales;
GRANT SELECT ON Products TO Sales;
GRANT MEMBERSHIP IN GROUP Sales TO Sally;
GRANT MEMBERSHIP IN GROUP Sales TO Joe;
```

これで Joe と Sally はこれらのテーブルを使用するパーミッションを持ちますが、テーブルの所有者は Company であり、Sally と Joe は Company グループのメンバーではないので、彼らがテーブルを参照するときには、修飾を使用する必要があります。

```
SELECT *
FROM company.Customers;
```

この状況を変更するには、次のように Sales グループを Company グループのメンバーにします。

```
GRANT GROUP TO Company;
GRANT MEMBERSHIP IN GROUP Company TO Sales;
```

これで、Joe と Sally は、Sales グループのメンバーであると同時に間接的に Company グループのメンバーになり、修飾子なしでデータを参照できます。したがって、次の文を使えるようになります。

```
SELECT *
FROM Customers;
```

**注意**
Joe と Sally は company グループのメンバーシップ以外のパーミッションを持ちません。company グループには、明示的に付与されたテーブルパーミッションはありません (company ユーザー ID はテーブルの作成者であり、DBA 権限を持っているので、Salaries のようなテーブルを参照するパーミッションを暗黙のうちに持っています)。したがって、Joe と Sally が次のいずれかの文を実行するとエラーになります。

```
SELECT *
FROM Salaries;
SELECT *
FROM company.Salaries;
```

どちらの場合も、Joe と Sally は Salaries テーブルを参照するパーミッションを持っていません。

# 高度なセキュリティを実現するためのビューとプロシージャー

高レベルのセキュリティが必要なデータベースでは、テーブルに対して直接パーミッションを定義することには限界があります。ユーザーに与えたテーブルのパーミッションは、テーブル全体に対して適用されます。ところが、テーブルごとでなくユーザーのパーミッションをより厳密に定義する必要がある場合が、数多くあります。次に例を示します。

- employee テーブルにアクセスする必要があるユーザーに対して、テーブル内にある個人的な情報にまでアクセスを許可することは望ましくない。
- 営業担当者にセールスコールの詳細を含むテーブルの更新を許可したいが、担当者自身の部分に対するアクセスだけに制限したい。

これらのケースでは、会社のニーズに応じてパーミッションを調整するためにビューとストアドプロシージャーを使うことができます。この項では、パーミッション管理のためのビューとストアドプロシージャーの使い方について説明します。

**参照**

- 「ビュー」『SQL Anywhere サーバー SQL の使用法』
- 「ビューパーミッション」527 ページ

## セキュリティを調整するためのビュー

「ビュー」は、ベーステーブルから選択したローとカラムを含む、計算されたテーブルです。ビューは、ユーザーにテーブルの一部分だけに対するアクセス権を与える場合に便利です。その部分はローまたはカラムとして定義できます。たとえば、ユーザーに employee テーブルの

Salary カラムが見えないようにしたり、ユーザーが自分で作成したローだけを見られるようにしたりできます。

**例 1**

Sales Manager は、自分の部署の営業部員に関するデータベースの情報にアクセスする必要があります。他部署の従業員の情報にアクセスしなければならない理由はありません。

例として、Sales Manager のユーザー ID を作成してから必要な情報を得るためのビューを作成し、Sales Manager のユーザー ID に適切なパーミッションを与える手順を次に示します。

1. GRANT 文を使用して、新しいユーザー ID を作成します。DBA 権限のあるユーザーとしてログインして、次の文を実行します。

```
CONNECT DBA
IDENTIFIED by sql;

CREATE USER SalesManager
IDENTIFIED BY sales;
```

2. 営業部の従業員だけを見るビューを次のように定義します。

```
CREATE VIEW EmployeeSales AS
  SELECT EmployeeID, GivenName, Surname
  FROM Employees
  WHERE DepartmentID = 200;
```

   テーブル参照を所有者で修飾することによって、同じ名前のテーブルの参照と混同されるおそれがなくなります。

3. SalesManager にビューを見るパーミッションを与えます。

```
GRANT SELECT
ON EmployeeSales
TO SalesManager;
```

   まったく同じ文を使用して、ビューおよびテーブルに対するパーミッションを与えます。

**例 2**

次の例では、Sales Manager が注文のまとめを確認できるようにするビューを作成します。このビューは、複数のテーブルからの情報を必要とします。

1. ビューを作成します。

```
CREATE VIEW OrderSummary AS
  SELECT OrderDate, Region, SalesRepresentative, CompanyName
  FROM SalesOrders
    KEY JOIN Customers;
```

2. Sales Manager にこのビューを見るパーミッションを与えます。

```
GRANT SELECT
ON OrderSummary
TO SalesManager;
```

3. プロセスが正常に動作したことをチェックするには、ユーザー ID SalesManager に接続し、作成したビューを見ます。

```
CONNECT SalesManager
IDENTIFIED BY sales;
SELECT *
FROM DBA.EmployeeSales;
SELECT *
FROM DBA.OrderSummary;
```

Sales Manager には基本となるテーブルを見るパーミッションは与えられていません。次の文はパーミッションエラーを起こします。

```
SELECT * FROM GROUPO.Employees;
SELECT * FROM GROUPO.SalesOrders;
```

### ビューに対するその他のパーミッション

前述の例では、SELECT パーミッションを調整するためのビューの使用法を説明しました。同じ方法で、INSERT、DELETE、UPDATE の各パーミッションをビューに付与できます。

## セキュリティを調整するためのプロシージャー

ビューはデータへのアクセスを制限しますが、プロシージャーはユーザーの行動を制限します。ユーザーは、プロシージャーの対象になるテーブルに対するパーミッションがなくても、プロシージャーの EXECUTE パーミッションを持つことができます。「プロシージャパーミッション」527 ページを参照してください。

### 厳密なセキュリティ

セキュリティを完全にするには、基本となるテーブルへのアクセスをすべて禁止し、ユーザーまたはユーザーのグループには、特定のストアドプロシージャーを実行するパーミッションだけを与えます。この方法であれば、データベースのデータの修正方法を厳密に定義できます。

## ネストオブジェクトの所有権

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

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

- ビューを作成するには、ユーザーは、そのビューに含まれるすべてのベースオブジェクト (テーブルやビューなど) に対する SELECT パーミッションが必要です。
- ビューにアクセスするには、ビューの所有者は、基本のテーブルまたはビューに対する適切なパーミッションを GRANT オプションで付与されている必要があります。また、ユーザーは、ビューに対する適切なパーミッションを付与されている必要があります。
- WHERE 句を使用して更新するには、SELECT パーミッションと UPDATE パーミッションの両方が必要です。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

- ユーザーがビュー定義内のテーブルを所有している場合は、そのユーザーがビューの所有者ではなく、ビューに対するアクセス権を付与されていなくても、ビューを介してテーブルにアクセスできます。

ネストされたプロシージャーには、次の DAC 規則が適用されます。

- プロシージャーを作成する場合、ユーザーは、基本となるオブジェクト (たとえば、テーブル、ビュー、またはプロシージャー) に対するパーミッションを必要としません。
- プロシージャーを実行する場合、プロシージャーの所有者は、そのプロシージャーが参照するオブジェクトに対する適切なパーミッションを必要とします。
- プロシージャーによって参照されるすべてのテーブルをユーザーが所有する場合でも、プロシージャーに対する EXECUTE パーミッションを付与されていないかぎり、プロシージャーを実行してテーブルにアクセスすることはできません。

この動作については、次の例で説明します。

## 例 1：user1 が table1 を作成し、user2 が table1 に基づく view2 を作成する

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

## 例 2：user2 が table1 にアクセスする procedure2 を作成する

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

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

- user3 が view3 を介して table1 と table2 にアクセスできるのは、user1 が table1 のパーミッションを user3 に付与し、なおかつ user2 が table2 のパーミッションを user3 に付与した場合です。

- user3 が table1 のパーミッションを持っているが table2 のパーミッションを持っていない場合、user3 は view3 を使用できません。table1 のカラムからなるサブセットにもアクセスできません。

- user1 または user2 が view3 を使用できるのは、(a) user1 が table1 のパーミッションを grant オプションとともに user3 に付与し、(b) user2 が table2 のパーミッションを grant オプションとともに user3 に付与し、さらに (c) user3 が view3 のパーミッションを user1 または user2 に付与した場合です。

# データベースオプション

データベースオプションは、データベースの動作をさまざまな面から制御します。たとえば、次の目的でデータベースオプションを使用できます。

- **互換性**　SQL Anywhere データベースの動作をどの程度まで Adaptive Server Enterprise に近づけるかを指定し、SQL が SQL/2008 に準拠しない場合にエラーを発生させるかどうかを設定できます。「互換性オプション」584 ページを参照してください。

- **エラー処理**　ゼロ除算などのエラーが起こったときの処理方法を制御できます。

- **同時実行性とトランザクション**　同時実行性の程度と、COMMIT 動作の詳細を制御できます。

# データベースオプション

オプションは SET OPTION 文を使用して設定します。一般的な構文は、次のとおりです。

**SET** [ **EXISTING** ] [ **TEMPORARY** ] **OPTION**
[ *userid*. | **PUBLIC**. ]*option-name* = [ *option-value* ]

特定のユーザーまたはグループにのみオプションを設定するには、ユーザー ID かグループ名を指定できます。すべてのユーザーは PUBLIC グループに属します。ユーザー ID またはグループを指定しない場合、SET OPTION 文を発行した、現在ログインしているユーザー ID にオプション変更が適用されます。

どのようなオプションでも、ユーザー定義であるかどうかにかかわらず、パブリック設定がなければユーザー固有の値を割り当てることはできません。データベースサーバーでは、ユーザー定義オプションへの TEMPORARY 値の設定はサポートされていません。

たとえば、次の文では、オプション変更を発行したユーザーが DBA の場合、ユーザー DBA にその変更が適用されます。

```
SET OPTION blocking_timeout = 3;
```

次の文は、すべてのユーザーが属するユーザーグループである PUBLIC ユーザー ID に変更を適用します。この文を実行するには、DBA 権限が必要です。

```
SET OPTION PUBLIC.login_mode = 'Standard';
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

*option-value* を省略すると、指定したオプション設定がデータベースから削除されます。これが個人的なオプション設定の場合は、値は PUBLIC 設定に戻ります。TEMPORARY オプションを削除すると、オプション設定は永久設定に戻ります。

**警告**
カーソルからローをフェッチしている間のオプション設定の変更はサポートされていません。このような変更を行うと、結果の信頼性が損なわれる可能性があるからです。たとえば、カーソルからフェッチしている間に date_format 設定を変えると、結果セットのロー同士の日付フォーマットが異なってしまいます。ローをフェッチしている間にオプション設定を変更しないでください。

**注意**
トルコ語照合を使用しているデータベースや大文字と小文字が区別されるデータベースでは、オプション名の大文字と小文字が正しくないと、SYSOPTION のクエリや次のようなクエリでは、どのローとも一致しない可能性があります。

```
SELECT * FROM sa_conn_properties( ) WHERE propname = 'BLOCKING';
```

オプション名における大文字と小文字の正しい対応の詳細については、「アルファベット順のオプションリスト」589 ページを参照してください。

### ♦ データベースオプションと Sybase Central

1. データベースサーバーを開きます。
2. データベースを右クリックし、**[オプション]** をクリックします。
3. 値を編集します。

**ヒント**
**[データベースオプション]** ウィンドウからも、特定のユーザーとグループに対してデータベースオプションを設定できます (あるユーザーまたはグループに対してこのウィンドウを開いた場合、ウィンドウ名はそれぞれ **[ユーザーのオプション]**、または **[グループのオプション]** になります)。

データベース自体のオプションを設定する場合は、実際にデータベースの PUBLIC グループのオプションを設定します。これは、すべてのユーザーとグループが、PUBLIC からオプションの設定を継承するためです。

**参照**

- 「SET OPTION 文」『SQL Anywhere サーバー SQL リファレンス』

## データベースオプションのスコープと継続期間

オプションは、パブリック、ユーザー、テンポラリの 3 つのスコープレベルで設定できます。

テンポラリオプションは、ユーザー設定とパブリック設定より優先されます。ユーザーレベルのオプションは、パブリック設定より優先されます。現在のユーザーに対してユーザーレベルのオプションを設定すると、対応するテンポラリオプションも設定されます。

オプションの中には (COMMIT 動作など)、スコープがデータベース全体に及ぶものがあります。これらのオプションの設定には、DBA 権限が必要です。その他のオプション (isolation_level など) は、現在の接続のみに適用され、特別なパーミッションは必要ありません。

オプション設定への変更がいつ有効になるかは、オプションによって異なります。recovery_time などのグローバルオプションの変更は、次にデータベースを起動したときに有効になります。

現在の接続にのみ影響を与えるオプションは、通常すぐに有効になります。オプションの設定は、たとえばトランザクションの途中でも変更できます。ただし、カーソルが開いているときにオプションを変更すると、結果の信頼性が低くなる可能性があります。たとえば、カーソルが開いているときに date_format を変更しても、次のローのフォーマットは変わりません。カーソルが取り出される方法によっては、変更がユーザーに伝わるのは、いくつか後のローになる場合があります。

### パブリックオプションの設定

PUBLIC ユーザー ID に対してオプションを設定するには、DBA 権限が必要です。

PUBLIC ユーザー ID に対するオプションの値を変更すると、独自の数値を設定していない全ユーザーについて、そのオプションの永久値が変更されます。ただし、そのオプションがすでに PUBLIC ユーザー ID に設定されていないかぎり、個々のユーザー ID にオプション値は設定されません。

PUBLIC ユーザーにかぎり設定されたオプションの中には、変更された設定を CONNECTION_PROPERTY 関数を通じてユーザーが参照できなくても、既存の接続に対して即座に有効になるものがあります。このような例には、global_database_id オプションがあります。このような理由から、他のユーザーがデータベースに接続している間は PUBLIC-only オプションを変更しないでください。

### テンポラリオプションの設定

SET OPTION 文に TEMPORARY キーワードを追加すると、変更の適用期間を変更できます。通常、オプションの変更は永久的です。SET OPTION 文を使って明示的に変更されるまで変わりません。

SET TEMPORARY OPTION 文を実行した場合、新しいオプション値は現在の接続にかぎり、接続の期間のみ有効になります。この新しい設定はサーバーメモリにのみ保存され、SYSOPTION システムビューには反映されません。

SET TEMPORARY OPTION を使用して PUBLIC オプションを設定すると、データベースの実行中はその変更が継続します。データベースが停止した際に、PUBLIC ユーザー ID のテンポラリオプションはその永久値に戻ります。

PUBLIC ユーザー ID のテンポラリオプションを設定すると、セキュリティ上の利点があります。たとえば、login_mode オプションが有効なときは、データベースは実行中のシステムのログインセキュリティに依存します。このオプション設定を一時的に有効にすると、Windows ドメインのセキュリティに依存しているデータベースは、データベースが停止し、ローカルコンピューター

にコピーされるといった場合でも、危険にさらされることはありません。この場合、login_mode オプションは、永久値の Standard に戻ります。このモードでは、統合化ログインを使用できません。

### SQL 文の設定オプション

INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT、INTERSECT 文には OPTION 句があるため、これらの文によってマテリアライズドビューが使用される方法や、クエリの最適化方法を指定できます。この句を使用すると、該当する文に対してのみ、パブリックオプションやテンポラリオプションの設定よりも優先されるオプション設定を指定できます。OPTION 句では、次のオプションの設定を変更できます。

- isolation_level
- max_query_tasks
- optimization_goal
- optimization_level
- optimization_workload
- user_estimates

### 参照


- 「INSERT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「UPDATE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「DELETE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「SELECT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「UNION 文」『SQL Anywhere サーバー SQL リファレンス』
- 「EXCEPT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「INTERSECT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「isolation_level オプション」630 ページ
- 「max_query_tasks オプション」644 ページ
- 「optimization_goal オプション」655 ページ
- 「optimization_level オプション」657 ページ
- 「optimization_workload オプション」658 ページ
- 「user_estimates オプション」706 ページ


## オプション設定の検索

さまざまな方法でオプション設定のリストや個別のオプションの値を入手できます。

### オプション値のリストを取得する

- システム定義オプションの現在の設定をすべてリストに表示するには、sa_conn_options システムプロシージャーを呼び出します。

```
CALL sa_conn_options;
```

このリストをアルファベット順に並べ替えるには、次の文を実行します。

```
SELECT *
FROM sa_conn_options( )
ORDER BY OptionName;
```

結果をフィルターするか、名前以外の要素で並べ替えるには、WHERE 句を使用します。次に例を示します。

```
SELECT *
FROM sa_conn_options( )
WHERE OptionDescription LIKE '%date%'
ORDER BY PropNum;
```

「sa_conn_options システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

- 現在の接続のシステム定義オプションの設定は、「接続プロパティ」のサブセットとしても取得できます。sa_conn_properties システムプロシージャーを使用すると、すべての接続プロパティをリストできます。

```
CALL sa_conn_properties;
```

このリストをアルファベット順に並べ替えるには、次の文を実行します。

```
SELECT *
FROM sa_conn_properties( )
ORDER BY PropName;
```

「sa_conn_properties システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

- sa_conn_options と sa_conn_properties システムプロシージャーの結果には、ユーザー設定または PUBLIC 設定よりも優先される、システム定義オプションの接続レベルの設定が含まれます。

ユーザー定義オプションの値は、sa_conn_options と sa_conn_properties システムプロシージャーのどちらからも返されません。代わりに、SYSOPTION または SYSOPTIONS カタログビューを直接問い合わせて、特定のユーザー定義オプションの値を判断できます。SYSOPTIONS システムビューで次のクエリを実行すると、永続的に設定されたすべてのシステム定義オプションとユーザー定義オプションについて、パブリックレベルとユーザーレベルの設定がすべて表示されます。

```
SELECT *
FROM SYSOPTIONS;
```

SET TEMPORARY OPTION 文を使用して特定の接続にサーバー定義オプションを一時的に設定しても、SYSOPTION (または SYSOPTIONS) システムビューは更新されません。接続レベルでオプション設定に加えた一時的な変更は、サーバーメモリにのみ保存されます。システム定義オプションの現在の設定を判断するには、sa_conn_options または sa_conn_properties システムプロシージャー、CONNECTION_PROPERTY 関数、または GET OPTION 文 (Embedded SQL の場合) を使用してください。

- Interactive SQL では、引数なしで SET 文を使用すると、システム定義接続、データベース、Interactive SQL のオプションの現在の設定がリストされます。

```
SET;
```

- Sybase Central でデータベースを選択し、**[ファイル]** » **[オプション]** をクリックします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 個別のオプション値を取得する

CONNECTION_PROPERTY システム関数を使用して、1 つのシステム定義オプションの現在の設定を取得できます。たとえば、次の文では、ansi_blanks オプションの値が表示されます。

```
SELECT CONNECTION_PROPERTY ( 'ansi_blanks' );
```

CONNECTION_PROPERTY を使用して、ユーザー定義オプションの値を取得することはできません。「CONNECTION_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

Embedded SQL プログラムでは、GET OPTION 文を使用して、Embedded SQL アプリケーション内のオプション (システム定義またはユーザー定義) の値を判断できます。「GET OPTION 文 [ESQL]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

### 参照

- 「SYSOPTION システムビュー」『SQL Anywhere サーバー SQL リファレンス』
- 「sa_conn_options システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「CONNECTION_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』
- 「GET OPTION 文 [ESQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「SET 文 [T-SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「SET REMOTE OPTION 文 [SQL Remote]」『SQL Anywhere サーバー SQL リファレンス』

### オプション設定のモニタリング

sa_server_option システムプロシージャーを使用して、データベースオプションを設定しようとしたときにメッセージを送信するか、エラーを返すことをデータベースサーバーに指示できます。

OptionWatchList プロパティを使用して、モニターするオプションのリストを作成し、OptionWatchAction プロパティを使用して、モニターされているオプションを設定しようとしたときにデータベースサーバーが実行する必要があるアクションを指定します。

### 例

次のストアドプロシージャーを呼び出して、データベースオプションの automatic_timestamp、float_as_double、tsql_hex_constant をモニターするようデータベースサーバーに指示します。

```
CALL dbo.sa_server_option(
'OptionWatchList','automatic_timestamp,float_as_double,tsql_hex_constant' );
```

次のストアドプロシージャーを呼び出して、OptionWatchList プロパティで指定されたオプションを設定しようとしたときにエラーを返すようデータベースサーバーに指示します。

```
CALL dbo.sa_server_option( 'OptionWatchAction','ERROR' );
```

### 参照

- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- OptionWatchAction データベースプロパティ 800 ページ
- OptionWatchList データベースプロパティ 800 ページ

# オプションの初期設定

SQL Anywhere への接続には、TDS プロトコル (Sybase Open Client 接続と jConnect JDBC 接続) と SQL Anywhere プロトコル (ODBC と Embedded SQL) のどちらも使用できます。

TDS プロトコルと SQL Anywhere 固有のプロトコルの両方を使用する場合、ストアドプロシージャーを使用して初期設定を行うことができます。SQL Anywhere ではこの方法を使用して、デフォルトの Adaptive Server Enterprise の動作に合わせて Sybase Open Client 接続と jConnect 接続を設定しています。

最初の設定は、login_procedure オプションによって制御されます。このオプションは、ユーザーが接続したときに実行されるストアドプロシージャーを指定します。デフォルトの設定では、sp_login_environment システムプロシージャーが使用されます。この動作は、必要に応じて変更できます。

sp_login_environment は、接続が TDS を介して行われているかどうかをチェックします。そうである場合は、sp_tsql_environment プロシージャーを呼び出して、いくつかのオプションに現在の接続に合った新しいデフォルト値を設定します。

**参照**

- 「login_procedure オプション」636 ページ
- 「sp_login_environment システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「sp_tsql_environment システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』

# オプション設定の削除

*option-value* を省略すると、指定したオプション設定がデータベースから削除されます。これが個人的なオプション設定の場合は、値は PUBLIC 設定に戻ります。TEMPORARY オプションを削除すると、オプション設定は永久設定に戻ります。

たとえば、次の文は、ansi_blanks オプションをデフォルト値にリセットします。

```
SET OPTION ansi_blanks =;
```

**参照**

- 「SET OPTION 文」『SQL Anywhere サーバー SQL リファレンス』

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

# オプション分類

SQL Anywhere には、さまざまなオプションが用意されています。いくつかの一般的なクラスに分類すると便利です。オプションのクラスは次のとおりです。

- データベースオプション。「データベースオプション」575 ページを参照してください。
- 互換性オプション。「互換性オプション」584 ページを参照してください。
- 同期オプション。「同期オプション」587 ページを参照してください。
- SQL Remote オプション。「SQL Remote オプション」588 ページを参照してください。
- Interactive SQL オプション。「Interactive SQL オプション」884 ページを参照してください。

## データベースオプション

次のオプションでは、SQL Anywhere データベースサーバーの動作を設定できます。

| オプション | 値 | デフォルト |
|---|---|---|
| 「allow_nulls_by_default オプション」 | On、Off | On<br><br>Sybase Open Client 接続と jConnect 接続の場合は Off |
| 「allow_read_client_file オプション」 | On、Off | Off |
| 「allow_snapshot_isolation オプション」 | On、Off | Off |
| 「allow_write_client_file オプション」 | On、Off | Off |
| 「ansi_blanks オプション」 | On、Off | Off |
| 「ansi_close_cursors_on_rollback オプション」 | On、Off | Off |
| 「ansi_permissions オプション」 | On、Off | On |
| 「ansi_substring オプション」 | On、Off | On |
| 「ansi_update_constraints オプション」 | Off、Cursors、Strict | Cursors |
| 「ansinull オプション」 | On、Off | On |
| 「auditing オプション」 | On、Off | Off |
| 「auditing_options オプション」 | 予約 | 予約 |
| 「background_priority オプション [廃止予定]」 | On、Off | Off |
| 「blocking オプション」 | On、Off | On |

| オプション | 値 | デフォルト |
|---|---|---|
| 「blocking_others_timeout オプション」 | 整数 (ミリ秒単位) | 0 |
| 「blocking_timeout オプション」 | 整数 (ミリ秒単位) | 0 |
| 「chained オプション」 | On、Off | On<br><br>Sybase Open Client 接続と jConnect 接続の場合は Off |
| 「checkpoint_time オプション」 | 分 | 60 |
| 「cis_option オプション」 | 0, 7 | 0 |
| 「cis_rowset_size オプション」 | 整数 | 50 |
| 「close_on_endtrans オプション」 | On、Off | On<br><br>jConnect 接続の場合は Off |
| 「collect_statistics_on_dml_updates オプション」 | On、Off | On |
| 「conn_auditing オプション」 | On、Off | On |
| 「connection_authentication オプション」 | 文字列 | 空の文字列 |
| 「continue_after_raiserror オプション」 | On、Off | On |
| 「conversion_error オプション」 | On、Off | On |
| 「cooperative_commit_timeout オプション」 | 整数 (ミリ秒単位) | 250 |
| 「cooperative_commits オプション」 | On、Off | On |
| 「database_authentication」 | 文字列 | 空の文字列 |
| 「date_format オプション」 | 文字列 | YYYY-MM-DD |
| 「date_order オプション」 | MDY、YMD、DMY | YMD |
| 「debug_messages オプション」 | On、Off | Off |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 値 | デフォルト |
|---|---|---|
| 「dedicated_task オプション」 | On、Off | Off |
| 「default_dbspace オプション」 | 文字列 | 空の文字列 (システム DB 領域を使用) |
| 「default_timestamp_increment オプション」 | 1 ～ 1000000 マイクロ秒を表す整数 | 1 |
| 「delayed_commit_timeout オプション」 | 整数 (ミリ秒単位) | 500 |
| 「delayed_commits オプション」 | On、Off | Off |
| 「divide_by_zero_error オプション」 | On、Off | On |
| 「escape_character オプション」 | 予約 | 予約 |
| 「exclude_operators オプション」 | 予約 | 予約 |
| 「extended_join_syntax オプション」 | On、Off | On |
| 「fire_triggers オプション」 | On、Off | On |
| 「first_day_of_week オプション」 | 1, 2, 3, 4, 5, 6, 7 | 7 (1 週間は日曜日から始まる) |
| 「for_xml_null_treatment オプション」 | Empty、Omit | Omit |
| 「force_view_creation オプション」 | 予約 | 予約 |
| 「global_database_id オプション」 | 整数 | 2147483647 |
| 「http_connection_pool_basesize オプション」 | 整数 (0 ～ 1000) | 10 |
| 「http_connection_pool_timeout オプション」 | 整数 (1 ～ 86400 秒) | 60 |
| 「http_session_timeout オプション」 | 整数 | 30 |
| 「integrated_server_name オプション」 | 文字列 | NULL |

| オプション | 値 | デフォルト |
|---|---|---|
| 「isolation_level オプション」 | 0、1、2、3、Snapshot、Statement-snapshot、Readonly-statement-snapshot | 0 |
| 「java_class_path option」 | 文字列 | 空の文字列 |
| 「java_location オプション」 | 文字列 | 空の文字列 |
| 「java_vm_options オプション」 | 文字列 | 空の文字列 |
| 「log_deadlocks オプション」 | On、Off | Off |
| 「login_mode オプション」 | Standard、Integrated、Kerberos、Mixed (旧式) | Standard |
| 「login_procedure オプション」 | 文字列 | sp_login_environment |
| 「materialized_view_optimization オプション」 | Disabled、Fresh、Stale、*N* { Minute[s] \| Hour[s] \| Day[s] \| Week[s] \| Month[s] } | Stale |
| 「max_client_statements_cached オプション」 | 整数 | 10 |
| 「max_cursor_count オプション」 | 整数 | 50 |
| 「max_plans_cached オプション」 | 整数 | 20 |
| 「max_priority オプション」 | Critical、High、Above Normal、Normal、Below Normal、Low、Background | Normal |
| 「max_query_tasks オプション」 | 整数 | 0 |
| 「max_recursive_iterations オプション」 | 整数 | 100 |
| 「max_statement_count オプション」 | 0 以上の整数 | 50 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 値 | デフォルト |
|---|---|---|
| 「max_temp_space オプション」 | 整数 [k\|m\|g\|p] | 0 |
| 「min_password_length オプション」 | 0 以上の整数 | 0 文字 |
| 「nearest_century オプション」 | 整数 (0 ～ 100) | 50 |
| 「non_keywords オプション」 | 文字列 | 空の文字列 |
| 「odbc_describe_binary_as_varbinary」 | On、Off | Off |
| 「odbc_distinguish_char_and_varchar オプション」 | On、Off | Off |
| 「oem_string オプション」 | 文字列 (最大 128 バイト) | 空の文字列 |
| 「on_charset_conversion_failure オプション」 | Ignore、Warning、Error | Ignore |
| 「on_tsql_error オプション」 | Stop、Conditional、Continue | Conditional<br><br>jConnect 接続の場合は Continue |
| 「optimization_goal オプション」 | First-row または All-rows | All-rows |
| 「optimization_level オプション」 | 0-15 | 9 |
| 「optimization_workload オプション」 | Mixed、OLAP | Mixed |
| 「pinned_cursor_percent_of_cache オプション」 | 整数 (0 ～ 100) | 10 |
| 「post_login_procedure オプション」 | 文字列 | 空の文字列 |
| 「precision オプション」 | 整数 (1 ～ 127) | 30 |
| 「prefetch オプション」 | Off、Conditional、Always | Conditional |
| 「preserve_source_format オプション」 | On、Off | On |
| 「prevent_article_pkey_update オプション」 | On、Off | On |

| オプション | 値 | デフォルト |
|---|---|---|
| 「priority オプション」 | Critical、High、Above Normal、Normal、Below Normal、Low、Background | Normal |
| 「progress_messages オプション」 | Off、Formatted、Raw | Off |
| 「query_mem_timeout オプション」 | -1、0、正の整数 | -1 |
| 「quoted_identifier オプション」 | On、Off | On |
| 「read_past_deleted オプション」 | On、Off | On |
| 「recovery_time オプション」 | 整数 (分) | 2 |
| 「remote_idle_timeout オプション」 | 整数 (秒) | 15 |
| 「request_timeout オプション」 | 整数 (0 ～ 86400 秒) | 0 |
| 「reserved_keywords オプション」 | 文字列 | 空の文字列 |
| 「return_date_time_as_string オプション」 | On、Off | Off |
| 「rollback_on_deadlock」 | On、Off | On |
| 「row_counts オプション」 | On、Off | Off |
| 「scale オプション」 | precision データベースオプションに指定した値より小さい 0 ～ 127 の整数 | 6 |
| 「secure_feature_key」 | 文字列 | NULL |
| 「sort_collation オプション」 | Internal、collation_name、collation_id | Internal |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 値 | デフォルト |
| --- | --- | --- |
| 「sql_flagger_error_level オプション」 | Off、SQL:1992/Entry、SQL:1992/Intermediate、SQL:1992/Full、SQL:1999/Core、SQL:1999/Package、SQL:2003/Core、SQL:2003/Package、UltraLite | Off |
| 「sql_flagger_warning_level オプション」 | Off、SQL:1992/Entry、SQL:1992/Intermediate、SQL:1992/Full、SQL:1999/Core、SQL:1999/Package、SQL:2003/Core、SQL:2003/Package、UltraLite | Off |
| 「string_rtruncation オプション」 | On、Off | On |
| 「st_geometry_asbinary_format オプション」 | このオプションでサポートされている値のリストは、ST_Geometry タイプの ST_AsBinary メソッドで説明されているパラメーターのリストと同じです。ST_Geometry タイプの ST_AsBinary メソッド『SQL Anywhere サーバー 空間データサポート』を参照してください。 | WKB |

| オプション | 値 | デフォルト |
| --- | --- | --- |
| 「st_geometry_astext_format オプション」 | このオプションでサポートされている値のリストは、ST_Geometry タイプの ST_AsText メソッドで説明されているパラメーターのリストと同じです。ST_Geometry タイプの ST_AsText メソッド『SQL Anywhere サーバー 空間データサポート』を参照してください。 | WKT |
| 「st_geometry_asxml_format オプション」 | このオプションでサポートされている値のリストは、ST_Geometry タイプの ST_AsXML メソッドで説明されているパラメーターのリストと同じです。ST_Geometry タイプの ST_AsXML メソッド『SQL Anywhere サーバー 空間データサポート』を参照してください。 | XML |
| 「st_geometry_describe_type オプション」 | CHAR、NCHAR、BINARY | CHAR |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 値 | デフォルト |
|---|---|---|
| 「st_geometry_interpolation オプション」 | key=value ペアのセットで、key1=value1;key2=value2; のようにフォーマットされます。有効なキー：relative-tolerance-percent=*DOUBLE* absolute-tolerance=*DOUBLE* | デフォルト化動作の詳細については、「st_geometry_interpolation オプション」690 ページを参照してください。 |
| 「st_geometry_on_invalid オプション」 | | |
| 「subsume_row_locks オプション」 | On、Off | On |
| 「suppress_tds_debugging オプション」 | On、Off | Off |
| 「synchronize_mirror_on_commit オプション」 | On、Off | Off |
| 「tds_empty_string_is_null オプション」 | On、Off | Off |
| 「temp_space_limit_check オプション」 | On、Off | On |
| 「time_format オプション」 | 文字列 | HH:NN:SS.SSS |
| 「time_zone_adjustment オプション」 | 整数、引用符で囲まれた負の整数、先頭に + または - が付いた時刻 (時間と分) を表す文字列 (引用符で囲まれたもの) のいずれか | クライアントの接続タイプに応じて、クライアントまたはデータベースサーバーのタイムゾーンのいずれかによって設定される |
| 「timestamp_format オプション」 | 文字列 | YYYY-MM-DD HH:NN:SS.SSS |
| 「timestamp_with_time_zone_format オプション」 | 文字列 | YYYY-MM-DD HH:NN:SS.SSS+HH:NN |
| 「truncate_timestamp_values オプション」 | On、Off | Off |
| 「tsql_outer_joins オプション」 | On、Off | Off |

| オプション | 値 | デフォルト |
|---|---|---|
| 「tsql_variables オプション」 | On、Off | Off<br><br>Sybase Open Client 接続と jConnect 接続の場合は On |
| 「updatable_statement_isolation オプション」 | 0, 1, 2, 3 | 0 |
| 「update_statistics オプション」 | On、Off | On |
| 「user_estimates オプション」 | Enabled、Disabled、Override-Magic | Override-Magic |
| 「uuid_has_hyphens オプション」 | On、Off | Off |
| 「verify_password_function オプション」 | 文字列 | 空の文字列 |
| 「wait_for_commit オプション」 | On、Off | Off |
| 「webservice_namespace_host オプション」 | NULL、hostname-string | NULL |

# 互換性オプション

次のオプションを使用すると、SQL Anywhere の動作に Adaptive Server Enterprise との互換性を持たせたり、両方の旧バージョンの動作をサポートしたり、ISO SQL/2008 動作を使用可能にしたりできます。

Adaptive Server Enterprise との互換性をさらに高めるために、SQL Anywhere の SET OPTION 文の代わりに Transact-SQL の SET 文を使用して、いくつかのオプションを現在の接続の間だけ設定できます。「SET 文 [T-SQL]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

## デフォルトの設定値

これらのオプションのデフォルト設定は、Adaptive Server Enterprise のデフォルト設定と一部異なります。SQL Anywhere データベースと Adaptive Server Enterprise データベース間の互換性を確保するためには、この項に挙げた個々の互換性オプションを明示的に設定する必要があります。

Sybase Open Client または JDBC インターフェイスを使用して接続が行われた場合は、Adaptive Server Enterprise との互換性のために、現在の接続にいくつかのオプションが明示的に設定されています。「Sybase Open Client と JDBC の Adaptive Server Enterprise との接続の互換性を維持するためのオプション」585 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## Sybase Open Client と JDBC の Adaptive Server Enterprise との接続の互換性を維持するためのオプション

| オプション | 設定 |
|---|---|
| allow_nulls_by_default | Off |
| ansi_blanks | Off |
| ansi_substring | On |
| ansinull | On |
| chained | Off |
| continue_after_raiserror | On |
| escape_character | Off |
| on_tsql_error | jConnect 接続の場合は Continue |
| time_format | HH:NN:SS.SSS |
| timestamp_format | YYYY-MM-DD HH:NN:SS.SSS |
| tsql_outer_joins | Off |
| tsql_variables | On |

## Transact-SQL と SQL/2008 の互換性オプション

次の表は、互換性オプション、指定可能な値、デフォルト設定のリストです。

| オプション | 値 | デフォルト |
|---|---|---|
| 「allow_nulls_by_default オプション」 589 ページ | On、Off | On |
| 「ansi_blanks オプション」 592 ページ | On、Off | Off |
| 「ansi_close_cursors_on_rollback オプション」 593 ページ | On、Off | Off |
| 「ansi_permissions オプション」 594 ページ | On、Off | On |

| オプション | 値 | デフォルト |
|---|---|---|
| 「ansi_update_constraints オプション」596 ページ | Off、Cursors、Strict | Cursors |
| 「ansinull オプション」597 ページ | On、Off | On |
| 「chained オプション」603 ページ | On、Off | On |
| 「close_on_endtrans オプション」605 ページ | On、Off | On |
| 「continue_after_raiserror オプション」609 ページ | On、Off | On |
| 「conversion_error オプション」610 ページ | On、Off | On |
| 「date_format オプション」613 ページ | 文字列 | YYYY-MM-DD |
| 「date_order オプション」615 ページ | MDY、YMD、DMY | YMD |
| 「divide_by_zero_error オプション」622 ページ | On、Off | On |
| 「escape_character オプション」622 ページ | 予約 | 予約 |
| 「fire_triggers オプション」624 ページ | On、Off | On |
| 「isolation_level オプション」630 ページ | 0, 1, 2, 3 | 0 |
| 「nearest_century オプション」649 ページ | 整数 (0 ～ 100) | 50 |
| 「non_keywords オプション」650 ページ | 文字列 (カンマで区切られたキーワードリスト) | 空の文字列 (オフになっているキーワードはない) |
| 「on_tsql_error オプション」655 ページ | Stop、Conditional、Continue | Conditional |
| 「quoted_identifier オプション」668 ページ | On、Off | On |
| 「reserved_keywords オプション」673 ページ | 文字列 (カンマで区切られたキーワードリスト) | 空の文字列 (オフになっているキーワードはない) |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 値 | デフォルト |
|---|---|---|
| 「sql_flagger_error_level オプション」679 ページ | Off、SQL:1992/Entry、SQL:1992/Intermediate、SQL:1992/Full、SQL:1999/Core、SQL:1999/Package、SQL:2003/Core、SQL:2003/Package、UltraLite | Off |
| 「sql_flagger_warning_level オプション」680 ページ | Off、SQL:1992/Entry、SQL:1992/Intermediate、SQL:1992/Full、SQL:1999/Core、SQL:1999/Package、SQL:2003/Core、SQL:2003/Package、UltraLite | Off |
| 「string_rtruncation オプション」691 ページ | On、Off | On |
| 「time_format オプション」696 ページ | 文字列 | HH:NN:SS.SSS |
| 「timestamp_format オプション」698 ページ | 文字列 | YYYY-MM-DD HH:NN:SS.SSS |
| 「tsql_outer_joins オプション」703 ページ | On、Off | Off |
| 「tsql_variables オプション」703 ページ | On、Off | Off |

## 同期オプション

次のデータベースオプションは、Mobile Link 同期クライアントとして使用する SQL Anywhere データベースの設定に使用できます。

| オプション | 値 | デフォルト |
|---|---|---|
| 「default_timestamp_increment オプション」618 ページ | 1 ～ 1000000 マイクロ秒を表す整数 | 1 |

| オプション | 値 | デフォルト |
|---|---|---|
| 「delete_old_logs オプション [SQL Remote]」621 ページ | On、Off、Delay、*n* days (日数) | Off |
| 「prevent_article_pkey_update オプション」664 ページ | On、Off | On |
| 「truncate_timestamp_values オプション」702 ページ | On、Off | Off |

## SQL Remote オプション

SQL Remote のレプリケーション動作を制御するために、次のオプションが用意されています。

| オプション | 値 | デフォルト |
|---|---|---|
| 「blob_threshold オプション [SQL Remote]」600 ページ | 整数 (バイト) | 256 |
| 「compression オプション [SQL Remote]」607 ページ | -1 ～ 9 の整数 | 6 |
| 「delete_old_logs オプション [SQL Remote]」621 ページ | On、Off、Delay、*n* days (日数) | Off |
| 「external_remote_options [SQL Remote]」623 ページ | On、Off | Off |
| 「qualify_owners オプション [SQL Remote]」667 ページ | On、Off | On |
| 「quote_all_identifiers オプション [SQL Remote]」668 ページ | On、Off | Off |
| 「replication_error オプション [SQL Remote]」671 ページ | ストアドプロシージャー名 | (プロシージャーなし) |
| 「replication_error_piece オプション [SQL Remote]」671 ページ | ストアドプロシージャー名 | (プロシージャーなし) |
| 「save_remote_passwords オプション [SQL Remote]」676 ページ | On、Off | On |
| 「sr_date_format オプション [SQL Remote]」681 ページ | *日付文字列* | yyyy/mm/dd |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 値 | デフォルト |
|---|---|---|
| 「sr_time_format オプション [SQL Remote]」682 ページ | *時刻文字列* | hh:nn:ss.Ssssss |
| 「sr_timestamp_format [SQL Remote]」683 ページ | *タイムスタンプ文字列* | yyyy/mm/dd hh:nn:ss.Ssssss |
| 「sr_timestamp_with_time_zone_format [SQL Remote]」684 ページ | *タイムゾーン付きタイムスタンプ文字列* | yyyy/mm/dd hh:nn:ss.Ssssss +hh:nn |
| 「subscribe_by_remote オプション [SQL Remote]」692 ページ | On、Off | On |
| 「verify_all_columns オプション [SQL Remote]」707 ページ | On、Off | Off |
| 「verify_threshold オプション [SQL Remote]」711 ページ | 整数 (バイト) | 1000 |

# アルファベット順のオプションリスト

この項では、オプションをアルファベット順に説明します。

## allow_nulls_by_default オプション

NULL か NOT NULL かを指定しないで作成された新規カラムが、NULL 値を含むのを許可するかどうかを制御します。

### 指定可能な値

On、Off

### デフォルト

On

Sybase Open Client 接続と jConnect 接続の場合は Off

### 備考

allow_nulls_by_default オプションは、Transact-SQL との互換性を保つために実装されています。

### 参照

- 「Transact-SQL 互換性のオプション」『SQL Anywhere サーバー SQL の使用法』
- allow_nulls_by_default 接続プロパティ 713 ページ
- 「Sybase Open Client と jConnect 接続の特性」1443 ページ
- 「CREATE TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「ALTER TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「Transact-SQL 互換性のオプション」『SQL Anywhere サーバー SQL の使用法』
- 「Transact-SQL と互換性のあるテーブル」『SQL Anywhere サーバー SQL の使用法』
- 「互換性オプション」584 ページ
- 「sp_tsql_environment システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』

## allow_read_client_file オプション

クライアントコンピューター上のファイルの読み込みを許可するかどうかを制御します。

### 指定可能な値

On、Off

### デフォルト

Off

### スコープ

DBA 権限が必要です。

### 備考

READ_CLIENT_FILE 関数を使用する場合など、クライアントコンピューターからファイルを読み込む場合は、このオプションを有効にしておく必要があります。

### 参照

- 「クライアントコンピューター上のデータへのアクセス」『SQL Anywhere サーバー SQL の使用法』
- 「READ_CLIENT_FILE 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』
- 「READCLIENTFILE 権限」540 ページ
- 「LOAD TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「isql_allow_read_client_file オプション [Interactive SQL]」890 ページ
- 「allow_write_client_file オプション」592 ページ
- 「isql_allow_write_client_file オプション [Interactive SQL]」891 ページ
- 「クライアント側データセキュリティ」『SQL Anywhere サーバー SQL の使用法』
- allow_read_client_file 接続プロパティ 713 ページ

## allow_snapshot_isolation オプション

スナップショットアイソレーションを有効または無効にします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 指定可能な値

On、Off

### デフォルト

Off

### スコープ

PUBLIC グループのみに設定できます。DBA 権限が必要です。

### 備考

このオプションは、データベースのスナップショットアイソレーションを有効にするかどうかを指定します。このオプションを On に設定すると、トランザクションがスナップショットアイソレーションを使用した場合、データベースサーバーは更新されたローの元のバージョンをテンポラリファイルに記録します。

トランザクションが実行中である場合は、allow_snapshot_isolation オプションの設定を変更しても、新しい設定が直ちに有効になるわけではありません。このオプションの設定を Off から On に変更しても、その時点で実行中のトランザクションが完了した後でなければスナップショットは使用できません。このオプションの設定を On から Off に変更した場合は、未完了のスナップショットが完了してからバージョン情報の収集が停止され、新たなスナップショットは開始されません。

特定のデータベースについて現在のスナップショットアイソレーションの設定を確認するには、SnapshotIsolationState データベースプロパティの値を問い合わせます。

```
SELECT DB_PROPERTY ( 'SnapshotIsolationState' );
```

SnapshotIsolationState プロパティの値は、次のいずれかです。

- **On**　データベースでスナップショットアイソレーションが有効になっている。
- **Off**　データベースでスナップショットアイソレーションが無効になっている。
- **in_transition_to_on**　現在のトランザクションの完了後にスナップショットアイソレーションが有効となる。
- **in_transition_to_off**　現在のトランザクションの完了後にスナップショットアイソレーションが無効となる。

### 参照


- 「isolation_level オプション」630 ページ
- 「updatable_statement_isolation オプション」704 ページ
- 「スナップショットアイソレーション」『SQL Anywhere サーバー SQL の使用法』
- 「独立性レベルと一貫性」『SQL Anywhere サーバー SQL の使用法』
- 「スナップショットアイソレーションの有効化」『SQL Anywhere サーバー SQL の使用法』
- allow_snapshot_isolation 接続プロパティ714 ページ


### 例

次の文は、データベースのスナップショットアイソレーションを有効にします。

```
SET OPTION PUBLIC.allow_snapshot_isolation = 'On';
```

## allow_write_client_file オプション

クライアントコンピューターへのファイルの書き込みを許可するかどうかを制御します。

### 指定可能な値

On、Off

### デフォルト

Off

### スコープ

DBA 権限が必要です。

### 備考

WRITE_CLIENT_FILE 関数を使用する場合など、クライアントコンピューターにファイルを書き込む場合は、このオプションを有効にしておく必要があります。

### 参照

- 「クライアントコンピューター上のデータへのアクセス」『SQL Anywhere サーバー SQL の使用法』
- 「WRITE_CLIENT_FILE 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』
- 「WRITECLIENTFILE 権限」541 ページ
- 「UNLOAD 文」『SQL Anywhere サーバー SQL リファレンス』
- 「isql_allow_write_client_file オプション [Interactive SQL]」891 ページ
- 「allow_read_client_file オプション」590 ページ
- 「isql_allow_read_client_file オプション [Interactive SQL]」890 ページ
- 「クライアント側データセキュリティ」『SQL Anywhere サーバー SQL の使用法』
- allow_write_client_file 接続プロパティ714 ページ

## ansi_blanks オプション

文字データがクライアントサイドでトランケートされるときの動作を制御します。

### 指定可能な値

On、Off

### デフォルト

Off

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 備考

ansi_blanks オプションが効力を持つのは、データベースが文字列の比較で後続ブランクを無視し、フェッチした文字列を文字配列に埋め込む場合に限られます。これは、*N* の値が *M* 以上の場合に、データ型 CHAR(*N*) の値が C char(*M*) 変数に読み込まれるたびトランケーションエラーを強制的に発生させます。ansi_blanks が Off に設定されていると、トランケーションエラーは、少なくともブランク以外の文字がトランケートされた場合にのみ発生します。

Embedded SQL で ansi_blanks オプションが On に設定されている場合、データ型 DT_STRING の値を挿入するときには、sqllen フィールドを値が保存されているバッファーの長さ (最低でも値の長さに末尾の NULL 文字を加えた長さ) に設定する必要があります。ansi_blanks が Off の場合、長さは NULL 文字の位置でのみ決定されます。ansi_blanks オプションの値は、接続が確立されたときに決定されます。接続が確立された後で ansi_blanks オプションの値を変更しても、sqllen Embedded SQL の動作には影響しません。

ブランク埋め込みが有効になっているデータベースでは、このオプションの設定によって、フェッチする式が CHAR または NCHAR であり (VARCHAR または NVARCHAR ではない)、ホスト変数 char または nchar (VARCHAR または NVARCHAR ではない) に格納される場合に、トランケーション警告をクライアントに送信するかどうかが決まります。これらの条件が該当し、ホスト変数が小さすぎるために、フェッチされた式の最大長までブランクが埋め込まれることにより、変数に式を格納できなくなる場合は、トランケーション警告が発生し、フェッチされた式の最大長までブランクが埋め込まれた場合に式の格納に必要となる最小バイト数がインジケーターに挿入されます。式が CHAR(N) または NCHAR(N) である場合、返される値の文字セット変換と文字長セマンティクスが考慮され、インジケーターが N 以外の値に設定されることがあります。

### 参照


- 「互換性オプション」584 ページ
- ansi_blanks 接続プロパティ714 ページ
- 「Sybase Open Client と jConnect 接続の特性」1443 ページ
- 「sp_tsql_environment システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』


## ansi_close_cursors_on_rollback オプション

WITH HOLD 句で開いたカーソルを、ROLLBACK を実行するときに閉じるかどうかを制御します。

### 指定可能な値

On、Off

### デフォルト

Off

### 備考

ドラフトの SQL/3 標準では、トランザクションをロールバックするときにすべてのカーソルが閉じている必要があります。デフォルトでは、ロールバックした場合、SQL Anywhere は、WITH

HOLD 句なしで開いたカーソルだけを閉じます。このオプションを使うと、すべてのカーソルを強制的に閉じることができます。

close_on_endtrans オプションは、ansi_close_cursors_on_rollback オプションを上書きします。

### 参照


- 「close_on_endtrans オプション」605 ページ
- 「互換性オプション」584 ページ
- ansi_close_cursors_on_rollback 接続プロパティ714 ページ
- 「カーソルとトランザクション」『SQL Anywhere サーバー プログラミング』


## ansi_permissions オプション

DELETE と UPDATE 文のパーミッションのチェックを制御します。

### 指定可能な値

On、Off

### デフォルト

On

### スコープ

PUBLIC グループのみに設定できます。すぐに有効になります。DBA 権限が必要です。

### 備考

ansi_permissions を On にすると、DELETE 文と UPDATE 文に対する SQL/2008 パーミッション要求がチェックされます。Adaptive Server Enterprise では、このオプションのデフォルト値は Off です。次の表はこの違いを説明しています。

| SQL 文 | ansi_permissions が OFF の時に必要なパーミッション | ansi_permissions が ON の時に必要なパーミッション |
|---|---|---|
| UPDATE | 値が設定されているカラムでの UPDATE パーミッション | 値が設定されているカラムでの UPDATE パーミッション<br><br>WHERE 句に指定されたすべてのカラムでの SELECT パーミッション<br><br>SET 句の右側に指定されたすべてのカラムでの SELECT パーミッション |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| SQL 文 | ansi_permissions が OFF の時に必要なパーミッション | ansi_permissions が ON の時に必要なパーミッション |
|---|---|---|
| DELETE | テーブルでは DELETE パーミッション | テーブルでは DELETE パーミッション<br><br>WHERE 句に指定されたすべてのカラムでの SELECT パーミッション |

ansi_permissions オプションは、PUBLIC グループのみに設定できます。個人的な設定は許可されません。

### 参照

- 「互換性オプション」584 ページ
- ansi_permissions 接続プロパティ714 ページ
- 「SET 文 [T-SQL]」『SQL Anywhere サーバー SQL リファレンス』

## ansi_substring オプション

start パラメーターまたは length パラメーターに負の値が設定された場合の SUBSTRING (SUBSTR) 関数の動作を制御します。

### 指定可能な値

On、Off

### デフォルト

On

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

### 備考

ansi_substring オプションを On に設定した場合、SUBSTRING 関数は ANSI/ISO SQL/2008 と同じ動作をします。開始オフセットが負または 0 の場合は、文字列の左側が文字以外で埋められているかのように扱われ、このときに負の長さが指定されるとエラーになります。

このオプションを Off に設定すると、SUBSTRING 関数は SQL Anywhere の以前のリリースと同じ動作となります。つまり、負の開始オフセットは文字列の末尾からのオフセットを意味し、負の長さは開始オフセットから length 文字左側の位置で部分文字列が終わることを意味します。また、開始オフセット 0 を使用した場合、開始オフセット 1 と同じ結果となります。

このオプションの設定は、BYTE_SUBSTR 関数の動作には影響しません。SUBSTRING 関数では、開始オフセットを正でない値に設定したり負の長さを指定したりしないことをおすすめします。可能なかぎり、SUBSTRING 関数の代わりに LEFT 関数または RIGHT 関数を使用してください。

### 参照

- 「SUBSTRING 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』
- 「LEFT 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』
- 「RIGHT 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』
- 「互換性オプション」584 ページ
- ansi_substring 接続プロパティ714 ページ

### 例

次の例は、ansi_substring オプションの設定によって SUBSTRING 関数の戻り値がどのように変わるかを示しています。

```
SUBSTRING( 'abcdefgh',-2,4 );
  ansi_substring = Off ==> 'gh' // substring starts at second-last character
  ansi_substring = On  ==> 'a'  // takes the first 4 characters of
                           // ???abcdefgh and discards all ?
SUBSTRING( 'abcdefgh',4,-2 );
  ansi_substring = Off ==> 'cd'
  ansi_substring = On  ==> value -2 out of range for destination

SUBSTRING( 'abcdefgh',0,4 );
  ansi_substring = Off ==> 'abcd'
  ansi_substring = On  ==> 'abc'
```

## ansi_update_constraints オプション

更新可能な範囲を制御します。

### 指定可能な値

Off、Cursors、Strict

### デフォルト

Cursors

### 備考

SQL Anywhere には、ANSI SQL 規格では許可されていない更新が可能な言語拡張機能がいくつか含まれています。これらの言語拡張機能を利用して、強力かつ効率的に更新を行うことができます。ただし、予期しない動作が発生することがあります。ユーザーのアプリケーションがこれらの言語拡張機能の動作を予期して設計されていない場合、この動作によって、更新内容の消失などの異常事態が発生することがあります。

ansi_update_constraints では、更新を SQL/2008 規格で許可される内容に限定するかどうかを制御します。

このオプションを Strict に設定すると、次の更新は許可されません。

●JOINS を含んだカーソルの更新

●ORDER BY 句に表示されるカラムの更新

●FROM 句は、UPDATE 文では使用できません。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

このオプションを Cursors に設定した場合は、カーソルにだけ同じ制限が加えられます。カーソルが FOR UPDATE または FOR READ ONLY で開かれていない場合は、データベースサーバーは SQL/2008 規格に基づいて、更新可能かどうかを選択します。ansi_update_constraints オプションが Cursors または Strict の場合、ORDER BY 句を含むカーソルは、デフォルトの FOR READ ONLY になります。それ以外の場合は、引き続きデフォルトの FOR UPDATE になります。

ODBC、JDBC、ADO.NET、OLE DB では、文の更新可能性は明示的な READ ONLY です。

Embedded SQL では、SQL プリプロセッサーの -m HISTORICAL オプションが指定されていないかぎり文の更新可能性は明示的な READ ONLY です。

デフォルトでは、ストアドプロシージャーのカーソルは明示的な FOR UPDATE または READ ONLY ではありません。

### 参照

- 「UPDATE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「SQL プリプロセッサー」『SQL Anywhere サーバー プログラミング』
- 「互換性オプション」584 ページ
- ansi_update_constraints 接続プロパティ714 ページ
- 「カーソルを使用したローの更新と削除」『SQL Anywhere サーバー プログラミング』

## ansinull オプション

NULL 値の解釈を制御します。

### 指定可能な値

On、Off

### デフォルト

On

### 備考

このオプションは、主に Transact-SQL (Adaptive Server Enterprise) との互換性を保つために実装されています。ansinull オプションは、NULL 定数を持つ比較述部の結果に影響します。また、NULL 値でグループ化されたクエリに対して発行される警告にも影響します。

ansinull を On にすると、ANSI 3 値的論理が WHERE 句、HAVING 句、または On 状態におけるすべての述部比較で使用されます。= または **!=** を使用した NULL との比較はすべて unknown と評価されます。

ansinull を Off に設定すると、SQL Anywhere は次の 4 つの条件に対して 2 値的論理を使用します。

*expr* = NULL

*expr* != NULL

*expr* = @var // @var はプロシージャー変数またはホスト変数

*expr* != @var

いずれの場合も、述部は true または false と評価され、unknown と評価されることはありません。このような比較では、NULL 値は各ドメインで特別値として処理され、2 つの NULL 値の等号 (=) 比較は true になります。式 *expr* は、相対的に単純な式で、カラム、変数、リテラルのみを参照し、サブクエリや関数を許可しない必要があります。

ansinull を On に設定した場合、NULL 値が少なくとも 1 つ含まれている式の集合関数の評価では、COUNT(*) を除き、警告 (SQLSTATE=01003) が生成されることがあります。 「集合関数では、NULL 値は無視されます。」『エラーメッセージ』を参照してください。

ansinull を Off に設定した場合、この警告は表示されません。

**注意**

- ansinull を Off に設定した場合、影響を受けるのは、SELECT 文、UPDATEDELETE 文、INSERT 文の述部である WHERE、HAVING、ON だけです。CASE 文、IF 文、または IF 式の比較のセマンティックは、影響を受けません。
- Adaptive Server Enterprise 12.5 では、ansinull が Off に設定された状態における NULL パターン文字列を持つ LIKE 述部の動作が変更されています。SQL Anywhere では、LIKE 述部は従来どおり ansinull の設定に影響を受けません。

**参照**


- 「互換性オプション」584 ページ
- 「sp_tsql_environment システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「SET 文 [T-SQL]」『SQL Anywhere サーバー SQL リファレンス』
- ansinull 接続プロパティ714 ページ
- 「Sybase Open Client と jConnect 接続の特性」1443 ページ


## auditing オプション

データベースでの監査を有効または無効にします。

### 指定可能な値

On、Off

### デフォルト

Off

### スコープ

PUBLIC グループのみに設定できます。すぐに有効になります。DBA 権限が必要です。

### 備考

このオプションは、監査のオンとオフを切り替えます。

監査とは、データベース内の多数のイベントに関する詳細をトランザクションログに記録することをいいます。監査を行うと、パフォーマンスに少し負荷がかかりますが、いくつかのセキュリティ機能を得られます。データベースの監査をオンにした場合、トランザクションログの使用を停止することはできません。トランザクションログを停止するためには、監査をオフにする必要があります。監査がオンであるデータベースを読み込み専用モードで起動することはできません。

auditing オプションを機能させるには、auditing オプションを On に設定し、sa_enable_auditing_type システムプロシージャーを使用して監査の対象とする情報のタイプを指定する必要があります。監査は、次のいずれかが該当する場合、有効になりません。

- auditing オプションが Off に設定されている
- 監査オプションが無効になっている

auditing オプションを On に設定し、監査オプションを指定しない場合は、すべてのタイプの監査情報が記録されます。または、sa_enable_auditing_type システムプロシージャーを使用して、パーミッションのチェック、接続試行、DDL 文、public オプション、トリガーから任意に組み合わせて記録することもできます。

### 参照


- 「データベースアクティビティの監査」1388 ページ
- 「sa_audit_string システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「sa_enable_auditing_type システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「sa_disable_auditing_type システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- auditing 接続プロパティ715 ページ


### 例

次の文は、データベースの監査をオンにします。

```
SET OPTION PUBLIC.auditing = 'On';
```

## auditing_options オプション

このオプションはシステムで使用するために予約されています。このオプションの設定は変更しないでください。

## background_priority オプション [廃止予定]

廃止予定。現在の接続以外の接続のパフォーマンスに対する影響を制限します。

### 指定可能な値

On、Off

### デフォルト

Off

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

このオプションを一時的に設定した場合、この設定は現在の接続にのみ適用されます。同じユーザー ID による別の接続では、このオプションの設定を変えることができます。

クエリ内並列処理は、background_priority が on に設定されている接続には使用されません。

### 備考

このオプションを On に設定すると、要求は Background 優先度レベルで実行されます。Off に設定すると、要求は priority オプションで指定した優先度レベルで実行されます。

### 参照


- 「priority オプション」664 ページ
- 「max_priority オプション」643 ページ
- background_priority 接続プロパティ716 ページ


## blob_threshold オプション [SQL Remote]

Message Agent がロングオブジェクト (BLOB) として処理する値のサイズを制御します。

### 指定可能な値

整数 (バイト)

### デフォルト

256

### 備考

blob_threshold オプションより長い値は、BLOB としてレプリケートされます。つまり、細かく分割して各部分をレプリケートしてから、SQL 変数を使用し、受信者サイトでその分割された部分を連結して再構成します。

SQL 文は、それぞれ 1 つのメッセージ内に収まる必要があります。したがって、このオプションの値をメッセージのサイズ (デフォルトでは 50 KB) より大きい値に設定しないでください。

### 参照


- 「SQL Remote オプション」『SQL Remote』
- 「BLOB」『SQL Remote』

## blocking オプション

ロック競合に応じる動作を制御します。

### 指定可能な値

On、Off

### デフォルト

On

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

### 備考

blocking オプションを On に設定すると、あるトランザクションが取得しようとしたロックが既存のロックと競合している場合、競合するすべてのロックが解放されるか、blocking_timeout に設定されたタイムアウト時間が経過するまで、そのトランザクションは待機状態となります。blocking_timeout ミリ秒以内にロックが解放されない場合は、待機しているトランザクションにエラーが返されます。blocking を Off に設定した場合は、トランザクションが競合するロックを取得しようとすると、エラーが返されます。

特定の操作 (マテリアライズドビューの再表示など) を行う間、blocking 接続プロパティの値を一時的に Off に設定できますが、それによってエラーが発生する場合があります。 「'%2' のローは、ユーザー '%1' によってロックされています。」『エラーメッセージ』を参照してください。

### 参照


- 「blocking_timeout オプション」602 ページ
- 「ブロックオプション」『SQL Anywhere サーバー SQL の使用法』
- blocking 接続プロパティ716 ページ


## blocking_others_timeout オプション

現在の接続がロールバックされる前に、現在の接続のローとテーブルのロックを他の接続からブロックできる時間を指定します。このオプションを使用すると、優先度の低いタスクが指定時間を超えて他の接続をブロックすることを防止できます。

### 指定可能な値

整数 (ミリ秒単位)

### デフォルト

0

### スコープ

Command Sequence 接続でのみサポートされます (TDS 接続や HTTP 接続またはイベントではサポートされません)。現在の接続の間、テンポラリオプションとしてのみ設定できます。

### 備考

このオプションを 0 に設定すると、現在の接続で他の接続を無制限にブロックできます。

このオプションで指定された時間 (ミリ秒) が経過しても現在の接続で別の接続がブロックされていると、現在の接続がロールバックされます。現在の接続が要求中である場合は、要求が中断されます。接続がロールバックされると、エラーが返されます。要求を中断する必要がある場合は、エラーが 2 回返されることがあります。接続がアイドルの場合は、すぐにロールバックされます。ロールバックされた場合、アプリケーションで失敗または遅延後の再試行を準備する必要があります。

**注意**
LOCK TABLE *table-name* WITH HOLD で作成したロックは、ロールバックされます。blocking_others_timeout エラーのために操作がロールバックされたことを示すエラーが接続で受信されると、その接続が切断されるまですべての要求でエラーが返されます。

### 参照

- blocking_others_timeout 接続プロパティ 716 ページ
- 「blocking_timeout オプション」602 ページ
- 「トランザクションのブロックとデッドロック」『SQL Anywhere サーバー SQL の使用法』
- 「LOAD TABLE 文」『SQL Anywhere サーバー SQL リファレンス』

## blocking_timeout オプション

トランザクションがロックを獲得するまで、どの程度の期間待機するかを制御します。

### 指定可能な値

整数 (ミリ秒単位)

### デフォルト

0

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

### 備考

blocking オプションを On に設定すると、トランザクションが取得しようとしたロックが既存のロックと競合している場合は、blocking_timeout に設定された時間 (ミリ秒単位) が経過するまで、競合するロックが解放されるのを待ちます。blocking_timeout ミリ秒以内にロックが解放されない場合は、待機しているトランザクションにエラーが返されます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

このオプションを 0 に設定すると、ロックを獲得しようとしているすべてのトランザクションは、競合するトランザクションがロックを解放するまで待機します。

### 参照

- 「blocking オプション」601 ページ
- blocking_others_timeout 接続プロパティ716 ページ
- 「request_timeout オプション」672 ページ
- 「デッドロック」『SQL Anywhere サーバー SQL の使用法』
- blocking_timeout 接続プロパティ716 ページ

## chained オプション

BEGIN TRANSACTION 文がない場合のトランザクションモードを制御します。

### 指定可能な値

On、Off

### デフォルト

On

Sybase Open Client 接続と jConnect 接続の場合は Off

### 備考

Transact-SQL トランザクションモードを制御します。非連鎖モード (chained = Off) では、明示的な BEGIN TRANSACTION 文が実行されてトランザクションを開始しないかぎり、各文が個々にコミットされます。連鎖モード (chained = On) では、データ検索文または修正文の前にトランザクションが暗黙的に開始されます。

### 参照

- 「互換性オプション」584 ページ
- 「sp_tsql_environment システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「オートコミット実装の詳細」『SQL Anywhere サーバー プログラミング』
- 「トランザクション」『SQL Anywhere サーバー SQL の使用法』
- chained 接続プロパティ717 ページ

## checkpoint_time オプション

データベースサーバーが、あるチェックポイントを実行してから次のチェックポイントを実行するまでの最大時間 (分単位) を設定します。

### 指定可能な値

整数

### デフォルト

60

### スコープ

PUBLIC グループのみに設定できます。DBA 権限が必要です。変更を有効にするには、データベースサーバーを停止し、再起動します。

### 備考

このオプションは、いつチェックポイントを実行すべきかを決定するために、recovery_time オプションと一緒に使用します。

### 参照


- 「チェックポイントログ」133 ページ
- 「recovery_time オプション」669 ページ
- 「データベースサーバーがチェックポイントのタイミングを決定する方法」1114 ページ
- 「-gc dbeng12/dbsrv12 サーバーオプション」232 ページ
- 「-m dbeng12/dbsrv12 サーバーオプション」255 ページ
- 「CHECKPOINT 文」『SQL Anywhere サーバー SQL リファレンス』
- checkpoint_time 接続プロパティ717 ページ


## cis_option オプション

リモートデータアクセス用のデバッグ情報をデータベースサーバーメッセージウィンドウに表示するかどうかを制御します。

### 指定可能な値

0, 7

### デフォルト

0

### スコープ

個々の接続または PUBLIC グループに設定できます。

### 備考

このオプションは、リモートデータアクセスを使用する場合に、リモートデータベースでのクエリの実行方法に関する情報をデータベースサーバーメッセージウィンドウに表示するかどうかを制御します。このオプションを 7 に設定すると、データベースサーバーメッセージウィンドウにデバッグ情報が表示されます。このオプションを 0 (デフォルト) に設定すると、データベースサーバーメッセージウィンドウにリモートデータアクセスのデバッグ情報は表示されません。

リモートトレーシングを有効にすると、データベースサーバーメッセージウィンドウにトレーシング情報が表示されます。この出力をファイルに記録するには、データベースサーバーの起動時に -o サーバーオプションを指定します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照


- 「-o dbeng12/dbsrv12 サーバーオプション」258 ページ
- 「接続のテスト」『SQL Anywhere サーバー SQL の使用法』
- cis_option 接続プロパティ718 ページ


## cis_rowset_size オプション

各フェッチに対してリモートサーバーから返されるローの数を設定します。

### 指定可能な値

整数

### デフォルト

50

### スコープ

個々の接続または PUBLIC グループに設定できます。リモートサーバーへの新しい接続が確立されたときに有効になります。

### 備考

このオプションは、ODBC を使用してリモートデータベースサーバーに接続する場合の ODBC FetchArraySize 値を設定します。

### 参照


- cis_rowset_size 接続プロパティ718 ページ


## close_on_endtrans オプション

トランザクションの終了時にカーソルを閉じるかどうかを制御します。

### 指定可能な値

On、Off

### デフォルト

On

jConnect 接続の場合は Off

### 備考

close_on_endtrans が On に設定されている場合、カーソルが WITH HOLD で開かれていないかぎり、カーソルはトランザクションがコミットされるたびに閉じます。トランザクションがロールバックするときの動作は ansi_close_cursors_on_rollback オプションの設定内容によって制御されます。

close_on_endtrans が Off に設定されている場合は、ansi_close_cursors_on_rollback オプションの設定に関係なく、またカーソルが WITH HOLD で開かれているかどうかにも関係なく、コミットまたはロールバックのどちらでもカーソルは閉じません。

このオプションを Off に設定すると、Adaptive Server Enterprise と互換性のある動作が得られます。

### 参照

- 「ansi_close_cursors_on_rollback オプション」593 ページ
- 「sp_tsql_environment システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「互換性オプション」584 ページ
- 「SET 文 [T-SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「Sybase Open Client と jConnect 接続の特性」1443 ページ
- close_on_endtrans 接続プロパティ719 ページ

## collect_statistics_on_dml_updates オプション

INSERT、DELETE、UPDATE などのデータ変更 DML 文の実行中に統計を収集するかどうかを制御します。

### 指定可能な値

On、Off

### デフォルト

On

### 備考

データベースサーバーは通常の文の実行中に統計情報を更新し、収集した統計を使用してカラム統計の自己チューニングを行います。collect_statistics_on_dml_updates オプションを Off に設定すると、INSERT、DELETE、UPDATE などのデータ変更 DML 文の実行中における統計の更新が無効となります。

通常の環境において、このオプションをオフにする必要はありません。ただし、きわめて大量のデータが頻繁に変更される環境では、このオプションを Off に設定することにより、パフォーマンスが向上することがあります。ただし、update_statistics も On に設定することが必要です。

collect_statistics_on_dml_updates オプションと update_statistics オプションが異なる点は、update_statistics オプションを On に設定した場合、ある述部を満たすローの実際の数とその述部を満たすと予想されるローの数が比較されてから、推定値が更新されることです。collect_statistics_on_dml_updates オプションを On に設定した場合は、挿入、更新、または削除されたローの値に基づいてカラム統計が修正されます。

### 参照

- 「update_statistics オプション」705 ページ
- 「カラム統計の更新」『SQL Anywhere サーバー SQL の使用法』
- collect_statistics_on_dml_updates 接続プロパティ719 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## compression オプション [SQL Remote]

SQL Remote メッセージの圧縮レベルを設定します。

### 指定可能な値

-1 ～ 9 の整数

### デフォルト

6

### 備考

値には次の意味があります。

- **-1**　メッセージをバージョン 5 フォーマットで送信します。バージョン 5 の Message Agent では、バージョン 6 以降の Message Agent で送信されたメッセージを読むことはできません。使用しているシステムのすべての Message Agent をバージョン 6 以降にアップグレードするまでは、compression オプションを -1 に設定してください。
- **0**　圧縮しません。
- **1 ～ 9**　圧縮率を高めます。大きい圧縮率でメッセージを作成すると、小さい圧縮率を使用した場合より時間がかかります。

### 参照

- 「SQL Remote オプション」『SQL Remote』
- 「メッセージサイズ」『SQL Remote』

## conn_auditing オプション

auditing オプションが On に設定されている場合に、接続ごとに監査を有効または無効にします。

### 指定可能な値

On、Off

### デフォルト

On

### スコープ

現在の接続の間、テンポラリオプションとしてのみ設定できます。DBA 権限が必要です。

### 備考

conn_auditing オプションの設定は、ログインプロシージャーに設定されている (login_procedure データベースオプションで指定します) 場合のみ有効です。conn_auditing を On に設定すると、その設定の監査がオンになります。ただし、auditing オプションも On に設定しないかぎり、監

査情報は記録されません。接続が監査されているかどうかを確認するには、次のプロシージャーを実行します。

```
SELECT CONNECTION_PROPERTY ( 'conn_auditing' );
```

### 参照


- 「監査の制御」1388 ページ
- 「auditing オプション」598 ページ
- 「login_procedure オプション」636 ページ
- conn_auditing 接続プロパティ720 ページ


## connection_authentication オプション

認証アプリケーションのデータベースシグネチャーに対する、アプリケーションシグネチャーの確認に使用する認証文字列を指定します。

### 指定可能な値

文字列

### デフォルト

空の文字列

### スコープ

個々の接続に対してのみ設定できます。

### 備考

このオプションは、SQL Anywhere データベースサーバーの OEM Edition を使用している場合にのみ有効になります。

認証アプリケーションは、接続確立後すぐに、すべての接続に対して connection_authentication データベースオプションを設定する必要があります。シグネチャーが検証されると、接続が認証され、SQL パーミッションで設定されている内容にかかわらず、アクティビティは制限されません。シグネチャーが検証されないと、接続は、非認証アプリケーションによって許可されているアクションに制限されます。

connection_authentication オプションは、TEMPORARY キーワードを使用して、現在の接続の間だけ設定できます。次の SQL 文は、接続を認証します。

```
SET TEMPORARY OPTION connection_authentication =
    'company = company-name;
    application = application-name;
    signature = application-signature';
```

*company-name* と *application-name* は、データベース認証の文と一致する必要があります。*application-signature* は、Sybase から取得するアプリケーションシグネチャーです。

会社名に引用符やアポストロフィなどの特殊文字が含まれている場合、受け入れられるようにするには、該当する文字を 2 つ続けて指定します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

SQL Anywhere の OEM Edition の構成と使用の詳細については、「認証 SQL Anywhere アプリケーション」181 ページを参照してください。

### 参照

- 「database_authentication」612 ページ
- connection_authentication 接続プロパティ720 ページ

### 例

次の例では、特殊文字を含む認証文字列を指定します。

```
SET TEMPORARY OPTION connection_authentication=
  'Company = Joe"s Garage;
   Application = Joe"s Program;
   Signature = 0fa55157edb8e14d818e...';
```

## continue_after_raiserror オプション

RAISERROR 文に応じて動作を制御します。

### 指定可能な値

On、Off

### デフォルト

On

### 備考

RAISERROR 文はプロシージャー内で使用され、エラー生成を実行します。このオプションを Off に設定すると、RAISERROR 文に到達するたびにプロシージャーまたはトリガーの実行が停止します。

continue_after_raiserror オプションを On に設定すると、RAISERROR 文は実行終了エラーの信号を送信しなくなります。代わりに、RAISERROR ステータスコードとメッセージが格納され、プロシージャーが完了すると直前の RAISERROR が返されます。RAISERROR を引き起こしたプロシージャーが他のプロシージャーから呼び出された場合、最も外側のプロシージャーが終了するまで RAISERROR は返されません。

中間レベルの RAISERROR のステータスとコードは、プロシージャーが終了すると失われます。リターン時に RAISERROR と一緒にエラーが生じた場合は、新しいエラー情報が戻され、RAISERROR 情報は失われます。アプリケーションは異なる実行ポイントで @@error グローバル変数を検査して、中間 RAISERROR ステータスを問い合わせることができます。

continue_after_raiserror オプションの設定は RAISERROR 文の後の動作を制御することを目的とし、on_tsql_error オプションが Conditional (デフォルト) に設定されている場合のみ使用します。on_tsql_error オプションを Stop または Continue に設定した場合、その設定は continue_after_raiserror の設定より優先されます。

### 参照

- 「on_tsql_error オプション」655 ページ
- 「互換性オプション」584 ページ
- 「RAISERROR 文」『SQL Anywhere サーバー SQL リファレンス』
- continue_after_raiserror 接続プロパティ720 ページ

## conversion_error オプション

データベースからデータをフェッチするときの、データ型変換障害のレポートを制御します。

### 指定可能な値

On、Off

### デフォルト

On

### 備考

このオプションは、データがデータベースからフェッチされるとき、またはデータベースに挿入されるときにデータ型変換障害が発生した場合、データベースがエラー (conversion_error が On) と警告 (conversion_error が Off) のどちらとしてレポートするかを制御します。

conversion_error を On に設定すると、SQLE_CONVERSION_ERROR エラーが生成されます。このオプションを Off に設定すると、警告 SQLE_CANNOT_CONVERT が生成されます。

変換エラーが警告のみでレポートされた場合、変換できなかった値の代わりに NULL 値が使用されます。Embedded SQL では、インジケーター変数はエラーを出したカラムに -2 を設定します。

### 参照

- 「インジケーター変数：変換エラーの処理」『SQL Anywhere サーバー プログラミング』
- 「インポート中の変換エラー」『SQL Anywhere サーバー SQL の使用法』
- 「互換性オプション」584 ページ
- conversion_error 接続プロパティ720 ページ

## cooperative_commit_timeout オプション

トランザクションログ内の COMMIT エントリがいつディスクに書き込まれるかを管理します。

> **注意**
> バージョン 11 の時点で、コミット動作が自動的にチューニングされるためこのオプション設定は無視されます。

### 指定可能な値

整数 (ミリ秒単位)

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### デフォルト

250

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

### 備考

このオプションは、cooperative_commits が On に設定されている場合にかぎり意味を持ちます。データベースサーバーは、ディスクに書き込む前に、指定されたミリ秒数だけ、他の接続がログのページを埋めるのを待ちます。デフォルト設定は 250 ミリ秒です。

### 参照


- 「cooperative_commits オプション」611 ページ
- 「delayed_commit_timeout オプション」619 ページ
- 「COMMIT 文」『SQL Anywhere サーバー SQL リファレンス』
- cooperative_commit_timeout 接続プロパティ721 ページ


## cooperative_commits オプション

コミットがいつディスクに書き込まれるかを制御します。

> **注意**
> バージョン 11 の時点で、コミット動作が自動的にチューニングされるためこのオプション設定は無視されます。

### 指定可能な値

On、Off

### デフォルト

On

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

### 備考

cooperative_commits を Off に設定した場合、COMMIT はデータベースサーバーで受信されたときにディスクに書き込まれ、その後アプリケーションは継続が許可されます。

cooperative_commits を On (デフォルト) に設定し、他にアクティブな接続がある場合は、データベースサーバーは COMMIT をすぐにはディスクに書き込みません。アプリケーションは cooperative_commit_timeout オプションで設定した最大長になるまで、他にそのページに含めるものがないか待ってから、COMMIT をディスクに書き込みます。

cooperative_commits を On に設定し、cooperative_commit_timeout の設定を大きくすると、ディスク I/O の数が減少することによってデータベースサーバーの全体的なスループットが向上しますが、各接続のターンアラウンドタイムが長くなります。

cooperative_commits と delayed_commits の両方を On に設定している場合、cooperative_commit_timeout に設定された時間がページの書き込みなしで経過すると、アプリケーションが (コミットが実行されたかのように) 再開され、残りの時間 (delayed_commit_timeout の値から cooperative_commit_timeout の値を差し引いた長さ) は delayed_commits 間隔として使用されます。その後、ページは満杯になっていなくてもディスクに書き込まれます。

### 参照

- 「delayed_commits オプション」620 ページ
- 「COMMIT 文」『SQL Anywhere サーバー SQL リファレンス』
- cooperative_commits 接続プロパティ721 ページ

## database_authentication

データベースの認証文字列を設定します。

### 指定可能な値

文字列

### デフォルト

空の文字列

### スコープ

PUBLIC グループのみに設定できます。このオプションを有効にするには、データベースを再起動する必要があります。

### 備考

このオプションは、SQL Anywhere データベースサーバーの OEM Edition を使用している場合にのみ有効になります。

データベースが認証されると、正しい認証シグネチャーを指定する接続だけが、データベースへの操作を実行できます。認証されない接続は、読み込み専用モードで操作します。認証データベースを使用する場合は、SQL Anywhere の OEM Edition を使用する必要があります。

データベースを認証するには、データベースの database_authentication オプションを設定します。

```
SET OPTION PUBLIC.database_authentication =
 'company = company-name;
 application = application-name;
 signature = database-signature';
```

*company-name* と *application-name* の引数は、シグネチャーの取得時に Sybase に提供した値です。*database-signature* は、Sybase から受け取ったデータベースシグネチャーです。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

会社名に引用符やアポストロフィなどの特殊文字が含まれている場合、受け入れられるようにするには、該当する文字を 2 つ続けて指定します。

データベースサーバーが認証データベースをロードするとき、データベースサーバーメッセージウィンドウに、認証された会社とアプリケーションについて説明するメッセージが表示されます。このメッセージを調べて、database_authentication オプションが有効になっているかどうかを確認できます。メッセージの形式は次のとおりです。

このデータベースは次の使用を目的としてライセンスされています : アプリケーション : application-name

会社 : company-name

SQL スクリプトファイルに認証の文を格納し、長いシグネチャーを繰り返し入力することを回避できます。認証の文をファイル *%SQLANY12%¥scripts¥authenticate.sql* に格納すると、この文は、データベースの作成、再構築、または更新を行うたびに適用されます。

SQL Anywhere の OEM Edition の構成と使用の詳細については、「認証 SQL Anywhere アプリケーション」181 ページを参照してください。

### 参照


- 「connection_authentication オプション」608 ページ
- database_authentication 接続プロパティ721 ページ


### 例

```
SET OPTION PUBLIC.database_authentication =
    'company = MyCompany;
    application = MySQLAnywhereApp;
    signature = 0fa55157edb8e14d818e';
```

## date_format オプション

データベースから取り出した日付のフォーマットを設定します。

### 指定可能な値

文字列

### デフォルト

YYYY-MM-DD

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

### 備考

フォーマットは次の記号を組み合わせた文字列です。

| 記号 | 説明 |
|---|---|
| YY | 2 桁の年 |
| YYYY | 4 桁の年 |
| MM | 2 桁の月 |
| MMM[m...] | 月を示す略式文字 |
| D | 曜日を示す 1 桁の数字 (1 = 日曜、7 = 土曜) |
| DD | 2 桁の日 |
| DDD[d...] | 曜日を示す略式文字 |
| JJJ | 通し日数 (1 ～ 366) |

各記号は、フォーマットされる日付のデータで置き換えられます。

日付フォーマットの解釈の制御については、「date_order オプション」615 ページを参照してください。

文字データがマルチバイトである場合、各記号の長さはバイト数ではなく文字数を表します。たとえば 'mmm' は 3 文字の月名を示しています。

文字データを表す記号 (*mmm* など) では、出力の文字を次のように制御できます。

- 記号をすべて大文字で入力すると、フォーマットはすべて大文字で表記されます。たとえば、MMM と入力すると、JAN と表記されます。
- 記号をすべて小文字で入力すると、フォーマットはすべて小文字で表記されます。たとえば、mmm と入力すると、jan と表記されます。
- 大文字と小文字を混ぜて入力すると、使用される言語に適切な文字が SQL Anywhere により選択されます。たとえば、Mmm と入力すると、英語では May、フランス語では mai と表記されます。

数値データを表す記号では、記号に大文字を使用するか小文字を使用するかで 0 埋め込みを制御できます。

- 記号をすべて大文字または小文字 (MM や mm など) で入力すると、0 埋め込みが行われます。たとえば、yyyy/mm/dd と入力すると、2002/01/01 と表記されます。
- 大文字と小文字を混ぜて入力すると (Mm など)、0 の埋め込みは行われません。たとえば、yyyy/Mm/Dd と入力すると、2002/1/1 と表記されます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**注意**
日付フォーマットの順序を変更するように date_format の設定を変更する場合は、同じ変更を反映するように date_order オプションも変更し、同様に date_order を変更する場合は date_format も変更してください。「date_order オプション」615 ページを参照してください。

### 参照


- 「日付を比較する探索条件」『SQL Anywhere サーバー SQL の使用法』
- 「time_format オプション」696 ページ
- 「timestamp_format オプション」698 ページ
- 「timestamp_with_time_zone_format オプション」700 ページ
- 「sp_tsql_environment システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- date_format 接続プロパティ721 ページ


### 例

次の表は、2008 年 4 月 14 日 (月) に実行された次の文からの出力と、date_format の設定を示します。

```
SELECT CAST( CURRENT DATE AS VARCHAR );
```

| date_format | CURRENT DATE |
|---|---|
| yyyy/mm/dd/ddd | 2008/04/14/mon |
| yyyy/Mm/Dd/ddd | 2008/4/14/mon |
| jjj | 105 |
| mmm yyyy | apr 2008 |
| Mmm yyyy | Apr 2008 |
| mm-yyyy | 04-2008 |

## date_order オプション

日付フォーマットの解釈を制御します。

### 指定可能な値

MDY、YMD、DMY

### デフォルト

YMD (ISO 日付フォーマット仕様に準拠)

Sybase Open Client 接続と jConnect 接続の場合、デフォルトは MDY に設定されます。

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

### 備考

データベースオプション date_order では、10/11/12 が Oct 11 1912、Nov 12 1910、Nov 10 1912 のうち、どの日付を意味するかを指定します。

> **注意**
> 日付フォーマットの順序を変更するように date_order の設定を変更する場合は、同じ変更を反映するように date_format オプションと timestamp_format オプションも変更し、同様に date_format または timestamp_format を変更する場合は date_order も変更してください。 「date_format オプション」613 ページと「timestamp_format オプション」698 ページを参照してください。

### 参照

- 「日付を比較する探索条件」『SQL Anywhere サーバー SQL の使用法』
- 「sp_tsql_environment システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- date_order 接続プロパティ721 ページ

## debug_messages オプション

DEBUG ONLY 句を含む MESSAGE 文を実行するか否かを制御します。

### 指定可能な値

On、Off

### デフォルト

Off

### 備考

このオプションにより、指定した DEBUG ONLY 句がある MESSAGE 文を含むストアドプロシージャーとトリガー内のデバッグメッセージの動作を制御できます。このオプションはデフォルトで Off に設定されており、MESSAGE 文が実行された場合にデバッグメッセージは表示されません。debug_messages を On に設定することにより、すべてのストアドプロシージャーとトリガーでデバッグメッセージが有効になります。

> **注意**
> debug_messages オプションを Off に設定している場合、DEBUG ONLY メッセージはパフォーマンスへの影響が小さいため、通常は運用システム上のストアドプロシージャーに残すことができます。ただし、頻繁に実行する位置では控え目に使用してください。そうしないと、パフォーマンスがわずかに低下することがあります。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照

- 「MESSAGE 文」『SQL Anywhere サーバー SQL リファレンス』
- debug_messages 接続プロパティ722 ページ

## dedicated_task オプション

要求処理タスクを、単一の接続からの要求処理に特化します。

### 指定可能な値

On、Off

### デフォルト

Off

### スコープ

現在の接続の間、テンポラリオプションとしてのみ設定できます。DBA 権限が必要です。

### 備考

dedicated_task 接続オプションを On に設定すると、要求処理タスクは、その接続の要求処理専用となります。このオプションを有効にして接続を事前に確立することで、応答しなくならないかぎりデータベースサーバーのステータスに関する情報を集められます。

### 参照

- dedicated_task 接続プロパティ722 ページ

## default_dbspace オプション

テーブルが作成されるデフォルト DB 領域を変更します。

### 指定可能な値

文字列

### デフォルト

空の文字列

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

### 備考

各データベースについて、システム (メイン) DB 領域に加えて最大 12 の DB 領域を作成できます。DB 領域を指定しないでテーブルを作成すると、このオプションに指定された DB 領域が使

用されます。このオプションを設定しない場合、空の文字列に設定した場合、または system に設定した場合は、システム DB 領域が使用されます。

テンポラリテーブルまたはインデックスは、default_dbspace オプションの設定に関係なく常に TEMPORARY DB 領域に配置されます。ベーステーブルの作成時に IN 句を指定した場合は、default_dbspace オプションで指定する DB 領域ではなく、IN 句で指定した DB 領域が使用されます。

すべてのテーブルをシステム DB 領域以外のロケーションに作成している場合は、システム DB 領域はチェックポイントログとシステムテーブルの保存にのみ使用されます。この設定は、パフォーマンス上の理由からチェックポイントログを他のデータベースオブジェクトとは別のディスクに保存する場合に便利です。チェックポイントログを別のディスクに保存するには、すべての CREATE TABLE 文で DB 領域を指定するか、このオプションの設定を変更してからテーブルを作成します。

### 参照


- 「追加の DB 領域の考慮事項」121 ページ
- 「異なるファイルの異なるデバイスへの配置」『SQL Anywhere サーバー SQL の使用法』
- 「CREATE DBSPACE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「ALTER DBSPACE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「DB 領域の作成」123 ページ
- 「-ds dbeng12/dbsrv12 データベースオプション」308 ページ
- default_dbspace 接続プロパティ722 ページ


### 例

次の例では、まず MyLibrary という名前の新しい DB 領域を作成します。次に、その DB 領域をデフォルト DB 領域に設定し、テーブル LibraryBooks をシステム DB 領域ではなく MyLibrary DB 領域に保存します。

```
CREATE DBSPACE MyLibrary
AS 'c:¥¥dbfiles¥¥library.db';
SET OPTION default_dbspace = 'MyLibrary';
CREATE TABLE LibraryBooks (
  title CHAR(100),
  author CHAR(50),
  isbn CHAR(30),
);
```

## default_timestamp_increment オプション

TIMESTAMP または UTC TIMESTAMP がデフォルト値になっているカラムを含むローが挿入または更新されるときに、このカラムの値を一意にしておくためにカラムに追加するマイクロ秒数を指定します。

### 指定可能な値

整数 (1 ～ 1000000)

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### デフォルト

1

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

### 備考

SQL Anywhere では、TIMESTAMP 値の精度が小数第 6 位であるため、2 つの同一の TIMESTAMP 値または TIMESTAMP WITH TIME ZONE 値を区別するために、デフォルトでは 1 マイクロ秒 (0.000001 秒) が追加されます。

DEFAULT TIMESTAMP 属性または DEFAULT UTC TIMESTAMP 属性を持つカラムを含む複数の行が更新され、カラム自体は明示的に更新されない場合は、更新される各ローの一意の値がカラムに与えられます。たとえば、DEFAULT UTC TIMESTAMP 属性を持つ TIMESTAMP WITH TIME ZONE タイプのカラムを含む 3 つのローが更新されるとします。これらのローは '2011-04-01 12:47:52.724000+00:00' に更新されますが、ローごとに一意の UTC TIMESTAMP 値が追加されます。

```
2011-04-01 12:47:52.724000+00:00
2011-04-01 12:47:52.724001+00:00
2011-04-01 12:47:52.724002+00:00
```

Microsoft Access などのソフトウェアの中には、TIMESTAMP 値を小数点第 3 位までトランケートしてしまうため、正しい比較を行う上で問題になるものもあります。互換性を確保するために、truncate_timestamp_values オプションを On に設定し、SQL Anywhere が格納する小数点以下の桁数を指定できます。

Mobile Link 同期に対し、このオプションを設定する場合は、最初の同期の実行前に設定する必要があります。

### 参照


- 「truncate_timestamp_values オプション」702 ページ
- 「TIMESTAMP 特別値」『SQL Anywhere サーバー SQL リファレンス』
- 「UTC TIMESTAMP 特別値」『SQL Anywhere サーバー SQL リファレンス』
- default_timestamp_increment 接続プロパティ722 ページ


## delayed_commit_timeout オプション

delayed_commits オプションが On のときに、アプリケーションが COMMIT を実行する時間から、COMMIT が実際にディスクに書き込まれる時間までの最大遅延を指定します。

### 指定可能な値

整数 (ミリ秒単位)

### デフォルト

500

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

### 備考

このオプションは、delayed_commits が On に設定されているときにのみ有効になります。このオプションは、トランザクションログの COMMIT エントリをどの時点でディスクに書き込むかを管理します。delayed_commits を On に設定すると、データベースサーバーは、delayed_commit_timeout オプションで指定されたミリ秒間待ち、他の接続によって 1 つのログページが満杯になってから、現在のページ内容をディスクに書き込みます。

### 参照


- 「delayed_commits オプション」620 ページ
- 「cooperative_commit_timeout オプション」610 ページ
- 「cooperative_commits オプション」611 ページ
- 「COMMIT 文」『SQL Anywhere サーバー SQL リファレンス』
- delayed_commit_timeout 接続プロパティ722 ページ


## delayed_commits オプション

COMMIT の後でデータベースサーバーからアプリケーションに制御が戻されるタイミングを決定します。

### 指定可能な値

On、Off

### デフォルト

Off (ISO の COMMIT 動作に準拠)

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

### 備考

On に設定すると、データベースサーバーは COMMIT のトランザクションログエントリがディスクに書き込まれるのを待たずに、直ちに COMMIT 文に応答します。Off に設定すると、アプリケーションは COMMIT がディスクに書き込まれるまで待たなければなりません。

このオプションが On の場合、ログページが満杯となったときと、delayed_commit_timeout オプションに設定された時間が経過したときのうち、いずれか早い時点でログがディスクに書き込まれます。トランザクションがコミットされた後であっても、データベースサーバーが COMMIT に応答してからページがディスクに書き込まれる前にシステム障害が発生すると、トランザクションが失われる可能性がわずかながらあります。delayed_commits を On に設定し、delayed_commit_timeout オプションを高い値にすると、応答時間は早くなりますが、リカバリ中にコミットされたトランザクションが失われるリスクが若干高くなります。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

cooperative_commits と delayed_commits の両方を On に設定している場合、cooperative_commit_timeout オプションに設定された時間がページへの書き込みなしで経過すると、アプリケーションが (コミットが実行されたかのように) 再開されます。残りの時間 (delayed_commit_timeout の値から cooperative_commit_timeout の値を差し引いた長さ) は delayed_commits 間隔として使用され、その時間が経過すると、ページは満杯になっていなくてもディスクに書き込まれます。

### 参照

- 「コミットの遅延の使用」『SQL Anywhere サーバー SQL の使用法』
- 「cooperative_commit_timeout オプション」610 ページ
- 「cooperative_commits オプション」611 ページ
- 「delayed_commit_timeout オプション」619 ページ
- 「COMMIT 文」『SQL Anywhere サーバー SQL リファレンス』
- delayed_commits 接続プロパティ722 ページ

## delete_old_logs オプション [SQL Remote]

トランザクションがレプリケートまたは同期されたときにトランザクションログを削除するかどうかを制御します。

### 指定可能な値

On、Off、Delay、*n* days (日数)

### デフォルト

Off

### 備考

このオプションは、SQL Anywhere Mobile Link クライアントと SQL Remote で使用されます。デフォルト設定は Off です。このオプションを On に設定すると、古いトランザクションログ内のすべての変更についてレプリケーションまたは同期が完了した時点で、古いトランザクションログが削除されます。Delay に設定すると、古いトランザクションログのファイル名に作成日が現在の日付であることが示されている場合、すべての変更が送信され、受信が確認されても、そのトランザクションログは削除されません。*n* days に設定すると、*n* 日より古いログが削除されます。無効な値は Off として解釈され、エラーや警告は報告されません。

delete_old_logs オプションを BACKUP 文と一緒に使用して古いトランザクションログのコピーを削除する方法については、「BACKUP 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

### 参照

- 「トランザクションログの管理」1108 ページ
- 「SQL Anywhere クライアントのログ」『Mobile Link クライアント管理』
- 「トランザクションログユーティリティ (dblog)」1041 ページ
- Mobile Link： 「MirrorLogDirectory (mld) 拡張オプション」『Mobile Link クライアント管理』
- SQL Remote： 「リモートデータベースのトランザクションログの管理」『SQL Remote』
- 「SQL Remote オプション」『SQL Remote』

### 例

delete_old_logs オプションが 10 日に設定されているリモートデータベースに対して、1 月 18 日に dbmlsync を実行すると、dbmlsync は、1 月 7 日以前に作成されたオフライントランザクションログを削除します。リモートデータベースは、オプションを次のように設定します。

```
SET OPTION delete_old_logs = '10 days';
```

## divide_by_zero_error オプション

ゼロ除算のレポートを制御します。

### 指定可能な値

On、Off

### デフォルト

On

### 備考

このオプションは、ゼロ除算をエラーとしてレポートするかどうかを指示します。このオプションを On に設定すると、ゼロ除算は SQLSTATE 22012 のエラーになります。

このオプションを Off に設定すると、ゼロ除算はエラーになりません。代わりに、NULL が返されます。

### 参照

- divide_by_zero_error 接続プロパティ723 ページ
- 「マテリアライズドビューの制限」『SQL Anywhere サーバー SQL の使用法』

## escape_character オプション

このオプションはシステムで使用するために予約されています。このオプションの設定は変更しないでください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## exclude_operators オプション

このオプションはシステムで使用するために予約されています。このオプションの設定は変更しないでください。

## extended_join_syntax オプション

複数テーブルのジョインに対して、重複する相関名構文を持つクエリを許可するか、またはエラーとしてレポートするかを指定します。

### 指定可能な値

On、Off

### デフォルト

On

### 備考

このオプションを On に設定すると、外部ジョインの NULL 入力側で重複した相関名を使用することが可能となります。同じ相関名で指定されたすべてのテーブルまたはビューは、テーブルまたはビューの同じインスタンスとして解釈されます。

次の FROM 句は、重複する相関名を使用したジョインが SQL Anywhere でどのように解釈されるかを示しています。C1 と C2 は探索条件を表します。

```
( R left outer join T on ( C1 ), T  join S on ( C2 ) )
```

このオプションを On に設定すると、このジョインは次のように解釈されます。

```
( R left outer join T on ( C1 ) ) join S on ( C2 )
```

このオプションを Off に設定すると、SQLCODE -137 が生成されます。「テーブル '%1' にはユニークな相関名が必要です。」『エラーメッセージ』を参照してください。

> **注意**
> 重複する相関名を削除した結果を確認する場合、2 番目の引数を ANSI に設定した REWRITE 関数を使うと、書き直した文を表示できます。

### 参照


- 「REWRITE 関数 [その他]」『SQL Anywhere サーバー SQL リファレンス』
- 「ジョインで重複する相関名 (スタージョイン)」『SQL Anywhere サーバー SQL の使用法』
- extended_join_syntax 接続プロパティ 724 ページ


## external_remote_options [SQL Remote]

メッセージリンクパラメーターを格納するかどうかを指定します。

### 指定可能な値

On、Off

### デフォルト

Off

### 備考

このオプションは SQL Remote が、メッセージリンクパラメーターをデータベースに格納するか (Off)、外部に格納するか (On) を指定するために使用します。

### 参照

- 「リモートメッセージタイプ制御パラメーターの設定」『SQL Remote』

## fire_triggers オプション

データベースにおけるトリガー起動を制御します。

### 指定可能な値

On、Off

### デフォルト

On

### 備考

On に設定すると、トリガーが起動されます。Off に設定すると、参照整合性トリガー (カスケード更新や削除など) を含め、トリガーは起動されません。DBA 権限を持つユーザーだけがこのオプションを設定できます。このオプションは -gf オプションによって上書きされます。このオプションを指定した場合、fire_triggers の設定にかかわらず、すべてのトリガー起動がオフになります。

Adaptive Server Enterprise トランザクションログのアクションは、トリガーによって実行されるアクションも含め、すべて SQL Anywhere にレプリケートされるので、このオプションは、Adaptive Server Enterprise から SQL Anywhere にデータをレプリケートする場合に意味を持ちます。

### 参照

- 「-gf dbeng12/dbsrv12 サーバーオプション」235 ページ
- 「トリガー」『SQL Anywhere サーバー SQL の使用法』
- 「互換性オプション」584 ページ
- fire_triggers 接続プロパティ724 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## first_day_of_week オプション

何曜日を週の最初にするかを設定します。

### 指定可能な値

1, 2, 3, 4, 5, 6, 7

### デフォルト

7 (1 週間は日曜日から始まる)

### 備考

値には次の意味があります。

| 値 | 意味 |
|---|---|
| 1 | 月曜日 |
| 2 | 火曜日 |
| 3 | 水曜日 |
| 4 | 木曜日 |
| 5 | 金曜日 |
| 6 | 土曜日 |
| 7 | 日曜日 |

このオプションで指定する値は、平日の値を取得するときに DATEPART 関数の結果に影響します。週の最初の曜日は、SET 文の DATEFIRST オプションを使用して変更することもできます。

このオプションで指定する値は、DOW 関数の結果には影響しません。たとえば、週の最初の曜日を月曜日に設定しても、DOW 関数は月曜日の値として 2 を返します。

### 参照

- 「DATEPART 関数 [日付と時刻]」『SQL Anywhere サーバー SQL リファレンス』
- 「DOW 関数 [日付と時刻]」『SQL Anywhere サーバー SQL リファレンス』
- 「SET 文 [T-SQL]」『SQL Anywhere サーバー SQL リファレンス』
- first_day_of_week 接続プロパティ 724 ページ

## for_xml_null_treatment オプション

FOR XML 句を使用するクエリでの NULL 値の処理を制御します。

### 指定可能な値

Empty、Omit

### デフォルト

Omit

### 備考

FOR XML 句を含むクエリを実行する場合、for_xml_null_treatment オプションが NULL 値の処理方法を決定します。デフォルトでは、NULL 値を含む要素と属性は、結果から省略されます。このオプションを Empty に設定すると、値が NULL の場合に空の要素または属性が生成されます。

### 参照

- 「クエリ結果を XML として取り出すための FOR XML 句の使用」『SQL Anywhere サーバー SQL の使用法』
- 「SELECT 文」『SQL Anywhere サーバー SQL リファレンス』
- for_xml_null_treatment 接続プロパティ 725 ページ

## force_view_creation オプション

このオプションはシステムで使用するために予約されています。このオプションの設定は変更しないでください。

**警告**
force_view_creation オプションは、*reload.sql* スクリプトでのみ使用されます。このオプションは、アンロードユーティリティ (dbunload) でのみ使用され、明示的に設定されません。

## global_database_id オプション

DEFAULT GLOBAL AUTOINCREMENT が指定されたカラムの値の範囲を設定します。レプリケーション環境でユニークなプライマリキーを生成する場合に使用します。

### 指定可能な値

負でない整数

### デフォルト

2147483647

### スコープ

PUBLIC グループのみに設定できます。DBA 権限が必要です。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 備考

このオプションで指定した値は開始値となります。DEFAULT GLOBAL AUTOINCREMENT が指定されたカラムでは、DEFAULT GLOBAL AUTOINCREMENT カラムに値がないテーブルにローを挿入すると、データベースサーバーによってそのカラムの値が自動的に生成されます。この値は、global_database_id の値とカラムの分割サイズによって決まります。

global_database_id をデフォルト値に設定すると、DEFAULT GLOBAL AUTOINCREMENT が無効になります。この場合、デフォルトとして NULL が生成されます。

現在のデータベースのオプション値を検索するには、次の文を使用します。

```
SELECT DB_PROPERTY( 'GlobalDBID' );
```

この機能は、特に、レプリケーション環境でユニークなプライマリキーを生成するために使用します。

### 参照

- 「CREATE TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「GLOBAL AUTOINCREMENT デフォルト」『SQL Anywhere サーバー SQL の使用法』
- 「AUTOINCREMENT カラムのあるテーブルの再ロード」『SQL Anywhere 12 変更点とアップグレード』
- global_database_id 接続プロパティ 725 ページ
- Mobile Link： 「グローバルデータベース ID の設定」『Mobile Link サーバー管理』
- SQL Remote： 「重複プライマリキーエラー」『SQL Remote』

### 例

次の例は、データベース ID 番号を 100 に設定します。

```
SET OPTION PUBLIC.global_database_id = '100';
```

## http_connection_pool_basesize オプション

データベース接続の公称スレッショルドサイズを指定します。

### 指定可能な値

整数 (0 ～ 1000)

### デフォルト

10

### 備考

このオプションの変更は、次のパージ循環時に 5 秒以内に有効になります。
http_connection_pool_basesize スレッショルド内の http_connection_pool_timeout を超過した未使用のデータベース接続は削除されます。接続プールのサイズが http_connection_pool_basesize スレッショルドを超えると、より積極的な早さで接続が削除されます。

値 0 を指定すると、オプションを設定してから 5 秒以内に HTTP 接続プールがパージされ、HTTP 接続プールが無効になります。

未使用の余分な接続はタイムアウト間隔の半分以内に削除され、残りの接続のうち半分は、タイムアウト間隔の 4 分の 1 以内に使用されない場合は削除される、のようになります。データベース内の接続プールの運用効率を判別するために、次のデータベースプロパティが追加されています。

- Http_Conn_Pool_Cached_Count
- Http_Conn_Pool_Hits
- Http_Conn_Pool_Misses
- Http_Conn_Pool_Steals

### 参照


- 「http_connection_pool_timeout オプション」628 ページ
- http_connection_pool_timeout 接続プロパティ726 ページ
- http_connection_pool_basesize 接続プロパティ726 ページ
- 「データベースプロパティ値のアクセス」781 ページ
- 「HTTP Web サービス」『SQL Anywhere サーバー プログラミング』


## http_connection_pool_timeout オプション

未使用の接続を接続プールに保持できる時間の上限を指定します。

### 指定可能な値

整数 (1 ～ 86400)

### デフォルト

60

### 備考

プールサイズが http_connection_pool_basesize を超えると、この値に基づいた増加率で、余分な接続が削除されます。

このオプションの変更は、次のパージ循環時に 5 秒以内に有効になります。
http_connection_pool_basesize スレッショルド内の http_connection_pool_timeout を超過した未使用のデータベース接続は削除されます。接続プールのサイズが http_connection_pool_basesize スレッショルドを超えると、より積極的な速さで接続が削除されます。

未使用の余分な接続はタイムアウト間隔の半分以内に削除され、残りの接続のうち半分は、タイムアウト間隔の 4 分の 1 以内に使用されない場合は削除される、のようになります。データベー

ス内の接続プールの運用効率を判別するために、次のデータベースプロパティが追加されています。

- Http_Conn_Pool_Cached_Count
- Http_Conn_Pool_Hits
- Http_Conn_Pool_Misses
- Http_Conn_Pool_Steals

### 参照


- 「http_connection_pool_basesize オプション」627 ページ
- http_connection_pool_basesize 接続プロパティ726 ページ
- http_connection_pool_timeout 接続プロパティ726 ページ
- 「データベースプロパティ値のアクセス」781 ページ
- 「HTTP Web サービス」『SQL Anywhere サーバー プログラミング』


## http_session_timeout オプション

非アクティブ状態の HTTP セッションが維持されるデフォルトのタイムアウト時間 (分単位) を指定します。

### 指定可能な値

整数 (1 ～ 525600)

### デフォルト

30

### スコープ

PUBLIC グループのみに設定できます。DBA 権限が必要です。

### 備考

このデータベースオプションを別の値に設定すると、以降に作成するすべての HTTP セッションに適用される新しいデフォルトのセッションタイムアウトが設定されます。sa_set_http_option システムプロシージャーを呼び出して特定の HTTP セッション内で SessionTimeout を設定すると、このデフォルトを HTTP セッションで上書きできます。

### 参照


- 「sa_set_http_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「HTTP サーバーでの HTTP セッションの管理」『SQL Anywhere サーバー プログラミング』
- SessionCreateTime 接続プロパティ743 ページ
- SessionTimeout 接続プロパティ744 ページ
- http_session_timeout 接続プロパティ726 ページ
- 「HTTP Web サービス」『SQL Anywhere サーバー プログラミング』

## integrated_server_name オプション

統合化ログインのために Windows ユーザーグループメンバーシップを検索するために使用するドメインコントローラーサーバーの名前を指定します。

### 指定可能な値

文字列

### デフォルト

NULL

### スコープ

PUBLIC グループのみに設定できます。DBA 権限が必要です。

### 備考

このオプションを使用すると、DBA 権限を持つユーザーは統合化ログインに Windows ユーザーグループを使う場合に、グループメンバーシップを検索するために使用するドメインコントローラーサーバーの名前を指定できます。デフォルトでは、グループメンバーシップの確認に SQL Anywhere データベースサーバーを実行中のコンピューターが使用されます。

### 参照


- 「Windows ユーザーグループの統合化ログイン」57 ページ
- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』
- integrated_server_name 接続プロパティ726 ページ


### 例

次の例では、グループメンバーシップがコンピューター server-1 で検証されることを指定します。

```
SET OPTION PUBLIC.integrated_server_name = '¥¥server-1';
```

## isolation_level オプション

ロック独立性レベルを制御します。

### 指定可能な値

0、1、2、3、Snapshot、Statement-snapshot、Readonly-statement-snapshot

### デフォルト

0

Sybase Open Client 接続、jConnect 接続、TDS 接続の場合は 1

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**備考**

このオプションは、次のようにロック独立性レベルを制御します。

- **0**　ダーティリード、繰り返し不可能読み出し、幻ローを許可します。
- **1**　ダーティリードを防ぎます。繰り返し不可能読み出しと幻ローを許可します。
- **2**　ダーティリードと繰り返し不可能読み出しを防ぎます。幻ローを許可します。
- **3**　直列化可能。ダーティリード、繰り返し不可能読み出し、幻ローを防ぎます。
- **Snapshot**　トランザクションが最初のローの読み込みまたは更新を行った時点から、コミットされたデータのスナップショットを使用します。
- **Statement-snapshot**　個々の文について、データベースから最初のローが読み込まれた時点から、コミットされたデータのスナップショットを使用します。繰り返し不可能読み出しと幻ローは、1 つのトランザクション内で発生することはありますが、1 つの文の中で発生することはありません。
- **Readonly-statement-snapshot**　読み込み専用の文についてのみ、データベースから最初のローが読み込まれた時点から、コミットされたデータのスナップショットを使用します。繰り返し不可能読み出しと幻ローは、1 つのトランザクション内で発生することはありますが、1 つの文の中で発生することはありません。更新可能な文については、updatable_statement_isolation オプションに指定された独立性レベル (0 (デフォルト)、1、2、3 のいずれか) を使用します。

このオプションを Snapshot、Statement-snapshot、または Readonly-statement-snapshot に設定する場合は、allow_snapshot_isolation を On に設定する必要があります。

SQL Anywhere JDBC ドライバーを使用する場合は、デフォルトの独立性レベルは 0 となります。

Snapshot、Statement-snapshot、Readonly-statement-snapshot のいずれかの独立性レベルでクエリを実行した場合には、データベースのコミットされた状態のスナップショットからデータが取得されます。

INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT、INTERSECT の各文に OPTION 句を含めることによって、各文で指定したオプション設定をこれらのオプションに対するテンポラリ設定やパブリック設定よりも優先させることができます。

### 参照

- 「独立性レベルと一貫性」『SQL Anywhere サーバー SQL の使用法』
- 「INSERT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「UPDATE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「DELETE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「SELECT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「UNION 文」『SQL Anywhere サーバー SQL リファレンス』
- 「EXCEPT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「INTERSECT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「sp_tsql_environment システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「allow_snapshot_isolation オプション」590 ページ
- 「updatable_statement_isolation オプション」704 ページ
- 「スナップショットアイソレーション」『SQL Anywhere サーバー SQL の使用法』
- 「独立性レベル選択のガイドライン」『SQL Anywhere サーバー SQL の使用法』
- 「互換性オプション」584 ページ
- 「BEGIN TRANSACTION 文 [T-SQL]」『SQL Anywhere サーバー SQL リファレンス』
- isolation_level 接続プロパティ727 ページ

## java_class_path option

クラスを検索する追加の一連のディレクトリまたは JAR ファイルを指定します。

### 指定可能な値

文字列

### デフォルト

空の文字列

### スコープ

PUBLIC グループのみに設定できます。DBA 権限が必要です。

### 備考

データベース内の Java で使用しているすべてのクラスと JAR ファイルはデータベース内にインストールすることをおすすめします。クラスと JAR ファイルをデータベース内に保存すると、データベースを簡単に別のコンピューターやオペレーティングシステムに移動できます。また、クラスと JAR ファイルをデータベース内にインストールすると、データベースサーバーのクラスローダーでクラスとリソースをデータベースからフェッチできるという利点もあります。その結果、データベース内の Java を使用している各接続で、これらのクラスの独自のインスタンスと、クラス内の静的変数の独自のコピーを使用できます。

ただし、クラスまたは JAR ファイルをシステムクラスローダーによってロードする必要がある場合は、java_class_path データベースオプションまたは -cp dbeng12/dbsrv12 サーバーオプションで指定できます。どちらのオプションを指定しても、Java VM の起動用にデータベースサーバーによって構築されるクラスパスにクラスと JAR ファイルが追加されます。java_class_path データベースオプションは、サーバーが複数のデータベースを実行していて、それぞれのデータベー

スにシステムクラスローダーでロードする必要のある異なるセットの JAR とディレクトリがあるときに便利です。-cp データベースサーバーオプションは、サーバー上のすべてのデータベースが同一のクラスまたは JAR ファイルを必要とするときに便利です。

### 参照

- 「-cp dbeng12/dbsrv12 サーバーオプション」213 ページ
- 「データベースにおける Java」『SQL Anywhere サーバー プログラミング』
- 「データベースに Java クラスを格納する方法は？」『SQL Anywhere サーバー プログラミング』
- 「java_class_path option」632 ページ

## java_location オプション

データベースの Java VM のパスを指定します。

### 指定可能な値

文字列

### デフォルト

空の文字列

### スコープ

PUBLIC グループのみに設定できます。DBA 権限が必要です。

### 備考

デフォルトでは、このオプションには空の文字列が設定されます。この場合、データベースサーバーは、JAVA_HOME 環境変数と Java VM のパスおよびその他のロケーションを検索します。java_location を設定していない場合にデータベースサーバーが使用する Java VM を確認するには、JavaVM データベースプロパティを使用します。

### 参照

- 「java_vm_options オプション」633 ページ
- 「Java VM」『SQL Anywhere サーバー プログラミング』
- JavaVM データベースプロパティ796 ページ
- java_location 接続プロパティ727 ページ

## java_vm_options オプション

Java VM の起動時にデータベースサーバーが使用するコマンドラインオプションを指定します。

### 指定可能な値

文字列

### デフォルト

空の文字列

### スコープ

PUBLIC グループのみに設定できます。DBA 権限が必要です。

### 備考

このオプションを使用すると、java_location オプションで指定された Java VM を起動するときにデータベースサーバーが使用するオプションを指定できます。これらの追加のオプションを使用して、デバッグ目的に Java VM を設定したり、UNIX プラットフォーム上のサービスとして実行したりできます。32 ビットモードではなく 64 ビットモードで Java VM を使用するために、その他のオプションが必要になることもあります。

### 参照


- 「java_location オプション」633 ページ
- 「Java VM」『SQL Anywhere サーバー プログラミング』
- java_vm_options 接続プロパティ727 ページ


### 例

次の例では、java_vm_options オプションを使用することで、データベースサーバーがサービスとして起動され、ユーザーがログアウトする必要が生じたときに、UNIX 上での Java VM の実行を維持しています。

```
SET OPTION PUBLIC.java_vm_options = '-Xrs';
```

次の例では、Java VM に対して HP-UX 上で 64 ビットモードで実行するように指示します。

```
SET OPTION PUBLIC.java_vm_options = '-d64';
```

## log_deadlocks オプション

デッドロックレポートのオン／オフを制御します。

### 指定可能な値

On、Off

### デフォルト

Off

### スコープ

PUBLIC グループのみに設定できます。DBA 権限が必要です。すぐに有効になります。

### 備考

このオプションを On に設定した場合、データベースサーバーは、内部バッファー内のデッドロックに関する情報をログに記録します。バッファーのサイズは、10000 バイトに固定されてい

ます。デッドロック情報は sa_report_deadlocks ストアドプロシージャーを使用すると表示できます。このオプションを Off に設定すると、バッファーの内容は保持されます。

デッドロックが発生すると、そのデッドロックに関わった接続のみの情報がレポートされます。接続のレポート順序は、どの接続がどの行を待っているかに基づきます。スレッドデッドロックの場合、すべての接続の情報がレポートされます。

デッドロックレポートがオンになっていると、デッドロックが発生した場合に Deadlock システムイベントを使用してアクションを実行することもできます。「システムイベント」1126 ページを参照してください。

sa_server_option システムプロシージャーで DeadlockLogging プロパティを使用して、このオプションの設定を変更することもできます。

### 参照

- 「sa_report_deadlocks システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「ブロックされているユーザーの判別」『SQL Anywhere サーバー SQL の使用法』
- 「チュートリアル：デッドロックの診断」『SQL Anywhere サーバー SQL の使用法』
- log_deadlocks 接続プロパティ729 ページ

## login_mode オプション

データベースの統合化ログインと Kerberos ログインの使用を制御します。

### 指定可能な値

Standard、Integrated、Kerberos、CloudAdmin、Mixed (旧式) のうち、1 つまたは複数

### デフォルト

Standard

### スコープ

PUBLIC グループのみに設定できます。DBA 権限が必要です。すぐに有効になります。

### 備考

このオプションは、標準ログイン、統合化ログイン、Kerberos ログイン、CloudAdmin ログインを許可するかどうかを指定します。以下のログインモードのうち 1 つまたは複数を指定できます (大文字と小文字は区別されません)。

- **Standard**　標準ログインを許可します。これがデフォルト設定です。標準ログインでは、ユーザー ID とパスワードの両方を入力する必要があり、Integrated や Kerberos の接続パラメーターは使用されません。
- **Integrated**　統合化ログインを許可します。
- **Kerberos**　Kerberos ログインを許可します。

- **CloudAdmin**　このログインモードは、クラウドの内部で使用します。
- **Mixed (旧式)**　Standard,Integrated の指定と同じです。

複数のログインモードを指定すると、データベースサーバーでは、そのすべてのモードが使用可能となります。

> **警告**
> login_mode データベースオプションを標準ログインが許可されない設定にした場合、接続できるのは、統合化ログインマッピングまたは Kerberos ログインマッピングを付与されているユーザーまたはグループだけに制限されます。ユーザー ID とパスワードで接続しようとすると、エラーが発生します。唯一の例外は、DBA 権限を持つユーザーです。

複数の値をカンマで区切ったリストとして指定できます。このリストに空白スペースを含めることはできません。たとえば、次の設定は標準ログインと統合化ログインの両方を許可します。

```
SET OPTION PUBLIC.login_mode = 'Standard,Integrated';
```

データベースファイルを保護していないため、権限のないユーザーがコピーできる場合は、一時的なパブリック login_mode オプションを使用する必要があります (統合化ログインと Kerberos ログインの両方で)。ファイルをコピーする場合、統合化ログインと Kerberos ログインはデフォルトではサポートされません。

### 参照


- 「Windows 統合ログイン」51 ページ
- 「Kerberos 認証」61 ページ
- 「データベースアクセス」1382 ページ
- 「セキュリティについての考慮事項：コピーされたデータベースファイル」74 ページ
- login_mode 接続プロパティ 729 ページ


### 例

次の文は、統合化ログインのみ有効にします (標準ログインと Kerberos ログインは失敗します)。

```
SET OPTION PUBLIC.login_mode = 'Integrated';
```

次の文は、標準ログインと Kerberos ログインを有効にします (統合化ログインは失敗します)。

```
SET OPTION PUBLIC.login_mode = 'Standard,Kerberos';
```

次の文は、標準ログイン、統合化ログイン、Kerberos ログインを有効にします。

```
SET OPTION PUBLIC.login_mode = 'Standard,Integrated,Kerberos';
```

## login_procedure オプション

起動時の接続互換性オプションを設定するログインプロシージャーを指定します。

### 指定可能な値

文字列


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### デフォルト

sp_login_environment システムプロシージャー

### スコープ

DBA 権限が必要です。

### 備考

このログインプロシージャーは、ランタイムに sp_login_environment プロシージャーを呼び出して、データベース接続設定を決定します。このログインプロシージャーは、すべてのチェックが完了し、接続が有効であることが確認された後で呼び出されます。login_procedure オプションで指定されたプロシージャーは、イベント接続では実行されませんが、Web サービス接続では実行されます。

新規プロシージャーを作成し、その新規プロシージャーを呼び出すための login_procedure を設定して、デフォルトデータベースオプション設定をカスタマイズできます。このカスタムプロシージャーは、sp_login_environment を呼び出すか、TDS 接続が確立されたことを検出し (デフォルトの sp_login_environment コードを確認します)、sp_tsql_environment を直接呼び出す必要があります。この操作が失敗した場合、TDS ベースの接続は切断されることがあります。sp_login_environment または sp_tsql_environment は編集しないでください。

ユーザー定義のログインプロシージャーを使用して、パスワードの有効期限切れを示す SQLSTATE 08WA0 のエラーメッセージを通知し、パスワードの有効期限が切れていることをユーザーに示すことができます。エラーの通知により、アプリケーションはエラーを確認し、有効期限の切れたパスワードを処理できます。パスワードの有効期限を実装する場合はログインポリシーを使用し、パスワードの有効期限切れのメッセージを返すログインプロシージャーは使用しないでください。

NewPassword=* 接続パラメーターを使用する場合は、このエラーを通知して、クライアントライブラリが新しいパスワードの入力を要求するプロンプトを表示できるようにする必要があります。プロシージャーで SQLSTATE 28000 (無効なユーザー ID またはパスワード) または SQLSTATE 08WA0 (パスワードの有効期限切れ) が通知されるか、RAISERROR のエラーが発生すると、ログインは失敗し、エラーがユーザーに返されます。その他のエラーを通知するか、別のエラーが発生した場合は、ユーザーログインは成功し、メッセージがデータベースサーバーメッセージログに書き込まれます。

### 参照


- 「post_login_procedure オプション」659 ページ
- 「sp_login_environment システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「sp_tsql_environment システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「パスワードのセキュリティ」1383 ページ
- 「NewPassword (NEWPWD) 接続パラメーター」355 ページ
- 「CREATE PROCEDURE 文」『SQL Anywhere サーバー SQL リファレンス』
- login_procedure 接続プロパティ729 ページ
- 「ログインポリシー」553 ページ

## 例

次に、INVALID_LOGON エラーを通知して接続を拒否するサンプルコードを示します。

```
CREATE PROCEDURE DBA.login_check( )
  BEGIN
    DECLARE INVALID_LOGON EXCEPTION FOR SQLSTATE '28000';
    // Allow a maximum of 3 concurrent connections
    IF( DB_PROPERTY( 'ConnCount' ) > 3 ) THEN
      SIGNAL INVALID_LOGON;
    ELSE
      CALL sp_login_environment;
    END IF;
  END
go

GRANT EXECUTE ON DBA.login_check TO PUBLIC
go

SET OPTION PUBLIC.login_procedure='DBA.login_check'
go
```

接続を禁止する代替方法の詳細については、「RAISERROR 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

次の例は、ユーザーの失敗した接続の数が -30 分間で 3 回よりも多くなった場合に、接続試行をブロックする方法を示しています。ブロック期間中にブロックされた試行は、すべて無効パスワードエラーを受け取り、ログに失敗として記録されます。DBA がログを解析するために、ログは十分な時間保持されます。

```
CREATE TABLE DBA.ConnectionFailure(
   pk INT PRIMARY KEY DEFAULT AUTOINCREMENT,
   user_name CHAR(128) NOT NULL,
   tm TIMESTAMP NOT NULL DEFAULT CURRENT TIMESTAMP
)
go

CREATE INDEX ConnFailTime ON DBA.ConnectionFailure(
   user_name, tm )
go

CREATE EVENT ConnFail TYPE ConnectFailed
HANDLER
BEGIN
   DECLARE usr CHAR(128);
   SET usr = event_parameter( 'User' );

   // Put a limit on the number of failures logged.
   IF (SELECT COUNT(*) FROM DBA.ConnectionFailure
      WHERE user_name = usr
      AND tm >= DATEADD( minute, -30,
         CURRENT TIMESTAMP )) < 20 THEN
      INSERT INTO DBA.ConnectionFailure( user_name )
         VALUES( usr );
      COMMIT;
      // Delete failures older than 7 days.
      DELETE DBA.ConnectionFailure
      WHERE user_name = usr
      AND tm < dateadd( day, -7, CURRENT TIMESTAMP );
      COMMIT;
   END IF;
END
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
go

CREATE PROCEDURE DBA.login_check( )
BEGIN
   DECLARE usr CHAR(128);
   DECLARE INVALID_LOGON EXCEPTION FOR SQLSTATE '28000';
   SET usr = CONNECTION_PROPERTY( 'Userid' );
   // Block connection attempts from this user
   // if 3 or more failed connection attempts have occurred
   // within the past 30 minutes.
   IF ( SELECT COUNT( * ) FROM DBA.ConnectionFailure
      WHERE user_name = usr
      AND tm >= DATEADD( minute, -30,
         CURRENT TIMESTAMP ) ) >= 3 THEN
      SIGNAL INVALID_LOGON;
   ELSE
      CALL sp_login_environment;
   END IF;
END
go

GRANT EXECUTE ON DBA.login_check TO PUBLIC
go

SET OPTION PUBLIC.login_procedure='DBA.login_check'
go
```

次の例は、ユーザーのパスワードの有効期限が切れていることを知らせるエラーを通知する方法を示しています。パスワードの有効期限切れの通知を実装するには、ログインポリシーを使用してください。

```
CREATE PROCEDURE DBA.check_expired_login( )
BEGIN
  DECLARE PASSWORD_EXPIRED EXCEPTION FOR SQLSTATE '08WA0';

  IF( condition-to-check-for-expired-password ) THEN
     SIGNAL PASSWORD_EXPIRED;
  ELSE
     CALL sp_login_environment;
  END IF;
END;
```

## materialized_view_optimization オプション

クエリへの応答を効率化するためにオプティマイザーがマテリアライズドビューを使用する方法を指定します。

### 指定可能な値

**Disabled**、**Fresh**、**Stale**、*N* { **Minute**[**s**] | **Hour**[**s**] | **Day**[**s**] | **Week**[**s**] | **Month**[**s**] }

### デフォルト

Stale

### スコープ

個々の接続、個々のユーザー、または PUBLIC グループに対して設定できます。すぐに有効になります。

### 備考

materialized_view_optimization オプションでは、オプティマイザーが古いマテリアライズドビューを使用できる状況を指定できます。

マテリアライズドビュー内のデータは、そのビューが参照するベーステーブルのデータが更新されることによって古くなります。マテリアライズドビューのリフレッシュ頻度を設定するときには、データの古さをどの程度まで許容できるかを検討する必要があります。また、リフレッシュプロセス中はビューがクエリに応答できないため、ビューのリフレッシュに要する時間も考慮に入れます。さらに、データベースの現在の状態を反映していない結果が返されることを受け入れられるかどうかについても考慮する必要があります。このオプションは、以下のいずれかに設定できます。

- **Disabled**　クエリの最適化にマテリアライズドビューを使用しません。
- **Fresh**　古くなっていない (基本となるテーブルが前回のリフレッシュ以降に変更されていない) 場合のみマテリアライズドビューを使用します。
- **Stale**　古くなっていてもマテリアライズドビューを使用します。これはデフォルト設定です。
- **N { Minute[s] | Hour[s] | Day[s] | Week[s] | Month[s] }**　古いマテリアライズドビューが指定された時間内にリフレッシュされている場合にかぎり、新しいマテリアライズドビューと古いマテリアライズドビューを使用します。指定する値は分単位とし、$2^{31}$ 分未満であることが必要です。データベースサーバーは、1 週間を 7 日、1 か月を 30 日と見なします。

クエリがマテリアライズドビューを直接参照する場合は、古くなっているかどうかにかかわらず、そのビューが使用されます。この場合、materialized_view_optimization オプションは効力を持ちません。

### 参照

- 「マテリアライズドビュー」『SQL Anywhere サーバー SQL の使用法』
- 「高度：マテリアライズドビューに対するデータの古さの制御の設定」『SQL Anywhere サーバー SQL の使用法』
- materialized_view_optimization 接続プロパティ 730 ページ

## max_client_statements_cached オプション

クライアントでキャッシュされる文の数を制御します。

### 指定可能な値

整数 (0 ～ 100)

### スコープ

個々の接続または PUBLIC グループに設定できます。値を変更すると、すぐに有効になります。

### デフォルト

10

### 備考

クライアントで文をキャッシュすると、同一の SQL 文が複数回準備されたときに、データベース要求と文の準備が減少します。同じ SQL 文の準備と削除が繰り返し行われると、クライアントは、文がデータベースサーバーに残された状態で、その文をキャッシュします。この文がアプリケーションによって削除された後でも、継続してキャッシュします。文のキャッシュにより、データベースサーバーは、文の削除や再準備などの余分な作業を節約できます。スキーマの変更、データベースオプション設定の変更、DROP VARIABLE 文の実行が生じると、準備文は自動的に削除され、その SQL 文が次に実行されるときに再び準備されます。これにより、不正な動作の原因となり得るような、キャッシュされた文が再使用されることのないようにします。

このオプションは、準備 (キャッシュ) されたまま維持できる文の最大数を指定します。キャッシュされた文は、max_statement_count リソースガバナーでは考慮されません。

このオプションの設定は、Embedded SQL、ODBC、OLE DB、ADO.NET、SQL Anywhere JDBC ドライバーを使用して作成された接続に適用されます。Sybase Open Client、jConnect、HTTP 接続には適用されません。

このオプションに 0 を設定すると、クライアントの文のキャッシュが無効になります。この値を増加すると、アプリケーションが同じ SQL 文について 10 回以上にわたって準備と削除を繰り返す場合に、パフォーマンスが向上する可能性があります。たとえば、25 の SQL 文をスルーするループを処理するアプリケーションについて考えてみます。ループをスルーするごとに準備と削除を繰り返し、これらの SQL 文のそれぞれにまったく同じ文が含まれているときは、このオプションを 25 に設定すると、パフォーマンスが向上します。

このオプションの値を増加させると、クライアントにおけるメモリの使用量が増加し、データベースサーバーに対するキャッシュ要求が高まります。キャッシュされた文が大量になった場合で、スキーマの変更やオプション設定の変更によって文を再使用できなくなると、その接続において、文のキャッシュは自動的に無効になります。文のキャッシュが自動的に無効にされると、クライアントは再び定期的に文のキャッシュを実行して処理方法を再評価し、文のキャッシュを再び有効にすることに効果があるかどうかを判断します。

クライアントでの文のキャッシュは、次の場合に予期しない結果をもたらす可能性があります。

1. 文を準備し、記述します。その記述は結果のない文を返します。

```
CREATE OR REPLACE FUNCTION test() RETURNS INT BEGIN RETURN 1;
  END;
  CALL test();
```

2. DDL 文によって同じ文のテキスト (この例では CALL 文) が同じ接続に結果セットを返すようになる場合。次に例を示します。

```
CREATE OR REPLACE PROCEDURE test() BEGIN SELECT 2;
```

```
END;
CALL test();
```

クライアントでの文のキャッシュが有効になっているときに、2 番目の CALL test() 文が誤って記述されます。

ログが *tracetime.pl* Perl スクリプトを使用して分析される場合は、要求ログが取得される間、max_client_statements_cached オプションを 0 に設定して、クライアントでの文のキャッシュを無効にする必要があります。

### 参照

- 「max_statement_count オプション」646 ページ
- ClientStmtCacheHits 接続プロパティ718 ページ
- ClientStmtCacheMisses 接続プロパティ719 ページ
- ClientStmtCacheHits サーバープロパティ756 ページ
- ClientStmtCacheMisses サーバープロパティ756 ページ
- max_client_statements_cached 接続プロパティ730 ページ

## max_cursor_count オプション

接続で一度に使用できるカーソルの最大数を制限するリソースガバナーを制御します。

### 指定可能な値

整数

### デフォルト

50

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。どの接続に対しても、このオプションを設定するには DBA 権限が必要です。

### 備考

このリソースガバナーを使用すると、接続ごとにユーザーが使用できるカーソルの数を、DBA が制限できます。接続の制限を超える操作が行われると、リソースのガバナーを超過していることを示すエラーを出力します。

接続がストアドプロシージャーを実行する場合、プロシージャーはプロシージャー所有者のパーミッションのもとで実行されます。ただし、プロシージャーが使用するリソースは、現在の接続に割り当てられています。

オプションを 0 (ゼロ) に設定すると、リソース制限を削除できます。

### 参照

- 「接続リソースの使用」552 ページ
- max_cursor_count 接続プロパティ730 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## max_plans_cached オプション

キャッシュに格納される実行プランの最大数を指定します。

### 指定可能な値

整数

### デフォルト

20

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。PUBLIC グループに対して、このオプションを設定するには DBA 権限が必要です。

### 備考

このオプションは、各接続でキャッシュされるプランの最大数を指定します。オプティマイザーは、ストアドプロシージャー、ファンクション、トリガーの中で実行されるクエリ、INSERT、UPDATE、DELETE の各文の実行プランをキャッシュします。ある接続でストアドプロシージャー、ストアド関数、またはトリガーに含まれる文が複数回実行された後、オプティマイザーは、その文の再使用可能なプランを構築します。

再利用可能なプランでは、選択性推定やリライト最適化にホスト変数の値は使用されません。この結果、文の最適化が再度行われた場合に比べ、再使用可能なプランのコストの方が高くつく可能性があります。再使用可能なプランのコストが文に最適と思われるコストに近いとき、オプティマイザーはそのプランをプランキャッシュに追加します。

このキャッシュは、CREATE TABLE や DROP TABLE など、テーブルスキーマを変更する文が実行されたときにクリアされます。宣言されたテンポラリテーブルを参照する文はキャッシュされません。

このオプションに 0 を設定すると、プランのキャッシュが無効になります。

### 参照

- 「プランのキャッシュ」『SQL Anywhere サーバー SQL の使用法』
- max_plans_cached 接続プロパティ 730 ページ

## max_priority オプション

接続の最高優先度レベルを制御します。

### 指定可能な値

Critical、High、Above Normal、Normal、Below Normal、Low、Background

### デフォルト

Normal

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。DBA 権限が必要です。

このオプションを一時的に設定した場合、この設定は現在の接続にのみ適用されます。同じユーザー ID による別の接続では、このオプションの設定を変えることができます。

### 備考

別の優先度レベルをスケジューリングすると、要求の優先度レベルに関係なく、すべての要求に CPU 時間が割り当てられます。優先度レベルが高い要求ほど、多くの時間が割り当てられます。

### 参照


- 「接続リソースの使用」552 ページ
- 「background_priority オプション [廃止予定]」599 ページ
- 「priority オプション」664 ページ
- max_priority 接続プロパティ 730 ページ


## max_query_tasks オプション

データベースサーバーがクエリの並列処理に使用できるサーバータスクの最大数を指定します。

### 指定可能な値

整数

### デフォルト

0

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

### 備考

max_query_tasks option は、SQL 文で使用できる並列処理の最大レベルを設定します。このオプションは、クエリの並列処理に使用できるデータベースサーバータスクの数を設定します。デフォルト値は 0 です。この場合、データベースサーバーは必要と判断した数のタスクを使用します。max_query_tasks オプションに 0 以外の値を指定した場合は、1 つのクエリについて使用可能なタスクの最大数が設定されます。max_query_tasks オプションを 1 に設定すると、クエリ内並列処理が無効になります。

サーバータスク、スレッド、クエリ実行の詳細については、「SQL Anywhere のスレッド化」161 ページと「マルチプログラミングレベルのデータベースサーバー設定」165 ページを参照してください。

すべての要求に対してデータベースサーバーが使用できるタスクの数は、起動時に -gn オプションで設定されたしきい値によって制限されます。この数は、サーバーがサービスを提供している

すべてのデータベースと接続に対してグローバルな最大数です。1 つの要求に使用されるタスクの数も、データベースサーバーで利用できる論理プロセッサーの数によって制限されます。たとえば、-gtc オプションでプロセッサーの同時実行性を 1 に設定した場合、クエリ内並列処理は無効となります。

クエリ内並列処理が有効である場合は、特定の条件を満たす SELECT 文がクエリ内並列処理によって処理されます。このクエリのアクセスプランでは、クエリ内並列処理が使用されたことが交換演算子によって示されます。

INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT、INTERSECT の各文に OPTION 句を含めることによって、各文で指定したオプション設定をこれらのオプションに対するテンポラリ設定やパブリック設定よりも優先させることができます。

### 参照


- 「-gn dbsrv12 サーバーオプション」237 ページ
- 「-gt dbeng12/dbsrv12 サーバーオプション」244 ページ
- 「-gtc dbeng12/dbsrv12 サーバーオプション」245 ページ
- 「INSERT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「UPDATE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「DELETE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「SELECT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「UNION 文」『SQL Anywhere サーバー SQL リファレンス』
- 「EXCEPT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「INTERSECT 文」『SQL Anywhere サーバー SQL リファレンス』
- max_query_tasks 接続プロパティ730 ページ


## max_recursive_iterations オプション

再帰共通テーブル式が反復できる最大回数を制限します。

### 指定可能な値

整数

### デフォルト

100

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。PUBLIC グループに対して、このオプションを設定するには DBA 権限が必要です。

### 備考

指定の反復回数内で計算が完了しなかった場合、再帰共通テーブル式の計算はアボートされてエラーが生成されます。再帰サブクエリは、反復が発生するたびに必要なリソースの総量が幾何学的に増加することがあります。このオプションを設定することによって、無限再帰が検出されるまでに消費される時間とリソースの量を制限し、しかも再帰共通テーブル式を意図した通りに動作させることができます。

このオプションに 0 を設定すると、再帰共通テーブル式が無効になります。

### 参照

- 「再帰共通テーブル式」『SQL Anywhere サーバー SQL の使用法』
- max_recursive_iterations 接続プロパティ730 ページ

## max_statement_count オプション

接続で一度に使用できる準備文の最大数を制限するリソースガバナーを制御します。

### 指定可能な値

整数

### デフォルト

50

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。どの接続に対しても、このオプションを設定するには DBA 権限が必要です。

### 備考

準備文を使用するアプリケーションは、準備文が不要になった後で明示的に削除されなかった場合、「準備文のリソースガバナーが制限を超えています。」というエラーを受け取ることがあります。max_statement_count オプションはリソースガバナーであり、これを使用することにより、DBA は 1 つの接続で使用される準備文の数を制限できます。接続の制限を超える操作が行われると、リソースのガバナーを超過していることを示すエラーを出力します。

接続がストアドプロシージャーを実行する場合、プロシージャーはプロシージャー所有者のパーミッションのもとで実行されます。ただし、プロシージャーが使用するリソースは、現在の接続に割り当てられています。

データベースサーバーには、接続によって作成される準備文ごとにデータ構造体が用意されます。アプリケーションがデータベースサーバーに準備文が不要になったことを通知するか、接続が切断されるまで、これらの構造体は解放されません。接続ごとの準備文の数を減らすためには、DROP STATEMENT 要求に相当する処理を実行する必要があります。次の表は、SQL Anywhere でサポートされる API で実行できる文を示しています。

| インターフェイス | 文 |
|---|---|
| ADO | RecordSet.Close |
| ADO.NET | SADataReader.Close または SADataReader.Dispose |
| Embedded SQL | DROP STATEMENT |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| インターフェイス | 文 |
|---|---|
| Java | resultSet.Close、Statement.Close |
| ODBC | SQLFreeStmt( hstmt, SQL_DROP ) または<br>SQLFreeHandle( SQL_HANDLE_STMT, hstmt ) |

**注意**
Java と .NET では、文を明示的に削除することをおすすめします。言語ルーチンは文リソースの割り付けを解除するサーバー呼び出しを発行しないため、ガーベジコレクションを使用して文を削除しないでください。また、ガーベジコレクションルーチンが実行される時点について保証はありません。

サーバーにおいて接続のいずれかの時点でデフォルトの最大数を超える準備文のサポートが必要になった場合は、max_statement_count をデフォルトより大きな値に設定する必要があります。ただし、アクティブな準備文が多くなると、サーバーメモリの使用量が増加します。
max_statement_count オプションを 0 (ゼロ) に設定し、準備文リソースガバナーを無効にすることもできますが、この方法は推奨できません。このようにすると、アプリケーションが準備文を適切に解放しない場合、データベースサーバーがメモリ不足によって異常終了しやすくなります。

### 参照


- 「接続リソースの使用」552 ページ
- 「準備文」『SQL Anywhere サーバー プログラミング』
- 「DROP STATEMENT 文 [ESQL]」『SQL Anywhere サーバー SQL リファレンス』
- max_statement_count 接続プロパティ731 ページ


## max_temp_space オプション

1 つの接続で使用できるテンポラリファイル領域の最大サイズを設定します。

### 指定可能な値

Integer [ **k** | **m** | **g** | **p** ]

### デフォルト

0

### スコープ

現在の接続の間、または PUBLIC グループのテンポラリオプションとして設定できます。すぐに有効になります。DBA 権限が必要です。

### 備考

このオプションでは、1 つの接続で使用できるテンポラリファイル領域の最大サイズを指定できます。このテンポラリファイル領域の最大サイズを超えると、要求は失敗します。

max_temp_space オプションが効力を持つためには、temp_space_limit_check オプションを On (デフォルト) に設定する必要があります。

デフォルト値の 0 は、1 つの接続が要求できるテンポラリファイル領域のサイズに制限がないことを意味します。0 以外の値は、1 つの接続で使用できるテンポラリファイル領域のバイト数を示します。単位をキロバイト、メガバイト、またはギガバイトで指定するには、それぞれ **k**、**m**、**g** を使用します。**p** を使用すると、値は使用できるテンポラリファイル領域の総量に対する割合を表します。

テンポラリファイル領域を要求する接続では、データベースサーバーが max_temp_space オプションの設定に対する制限をチェックし、要求がテンポラリファイル領域の最大サイズを超えていないことを確認します。接続が最大サイズを超えるテンポラリファイル領域を要求すると、その要求は失敗し、エラー SQLSTATE_TEMP_SPACE_LIMIT が生成されます。

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。PUBLIC グループに対して、このオプションを設定するには DBA 権限が必要です。

### 参照


- 「temp_space_limit_check オプション」695 ページ
- 「sa_disk_free_space システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- max_temp_space 接続プロパティ731 ページ


### 例

次の文は接続に対して 1 GB の制限を設定します。

```
SET OPTION PUBLIC.max_temp_space = '1g';
```

次の 2 つの文は、いずれも接続に対して 1 MB の制限を設定します。

```
SET OPTION PUBLIC.max_temp_space = 1048576;
```

```
SET OPTION PUBLIC.max_temp_space = '1m';
```

次の文は、テンポラリファイル領域の使用を全体の 5 % に制限します。

```
SET OPTION PUBLIC.max_temp_space = '5p';
```

## min_password_length オプション

新しいパスワードの最小長をデータベースに設定します。

### 指定可能な値

整数

値はバイト数で指定します。シングルバイト文字セットの場合、これは文字数と同じになります。

### デフォルト

0 文字

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### スコープ

PUBLIC グループのみに設定できます。すぐに有効になります。DBA 権限が必要です。

### 備考

このオプションを使用すると、データベース管理者は、セキュリティを強化するために、新しいパスワードすべてに最小長を設定できます。既存のパスワードは影響を受けません。パスワードは最大長が 255 バイトで、大文字と小文字が区別されます。

### 参照


- 「verify_password_function オプション」708 ページ
- 「パスワードのセキュリティ」1383 ページ
- min_password_length 接続プロパティ731 ページ


### 例

新しいパスワードの最小長を 6 バイトに設定します。

```
SET OPTION PUBLIC.min_password_length = 6;
```

## nearest_century オプション

文字列から日付への変換で、2 桁の年の解釈を制御します。

### 指定可能な値

整数 (0 ～ 100)

### デフォルト

50

### 備考

このオプションは、文字列から日付またはタイムスタンプに変換するときに、2 桁の年の処理を制御します。

nearest_century 設定は、ロールオーバーポイントとして動作する数値です。この値より小さい 2 桁の年は 20yy に変換され、この値以上の年は 19yy に変換されます。

従来の SQL Anywhere では、年に 1900 を加算していました。Adaptive Server Enterprise では最も近い世紀を使用するので、*yy* が 50 より小さい場合は 20yy になります。

### 参照


- 「互換性オプション」584 ページ
- nearest_century 接続プロパティ733 ページ

## non_keywords オプション

個々のキーワードをオフにして、識別子として使用できるようにします。

### 指定可能な値

文字列

### デフォルト

空の文字列

### 備考

このオプションは、個々のキーワードをオフにします。これにより、古いバージョンの製品で作成されたアプリケーションが新しいキーワードで破損されないことが保証されます。現在データベースにキーワードである識別子がある場合、すべてのアプリケーションまたはスクリプトで識別子を二重引用符で囲むか、non_keywords オプションを使用してキーワードをオフにできます。

キーワードセット、単一オプション、複数オプションをオフにすることはできません。単語がキーワードとして識別されるかどうかは、次の条件に従って判別されます (優先度順)。

- その単語が予約語の SQL Anywhere リストに表示される。
- その単語が reserved_keywords オプションでオンになっている。
- その単語が non_keywords オプションでオフになっている。

次の文を記述すると、TRUNCATE と SYNCHRONIZE がキーワードとして認識されません。

```
SET OPTION non_keywords = 'TRUNCATE, SYNCHRONIZE';
```

このオプションで新規設定を行うと、前に設定しているものに置き換わります。次の文は以前の設定内容をすべてクリアします。

```
SET OPTION non_keywords =;
```

このオプションを使用すると、オフにしたキーワードを使用する SQL 文を使えなくなります。そのような SQL 文を指定すると、構文エラーが生じます。

reserved_keywords オプションを使用して、個々のキーワードをオンにすることはできません。

### 参照


- 「reserved_keywords オプション」673 ページ
- 「キーワード」『SQL Anywhere サーバー SQL リファレンス』
- 「互換性オプション」584 ページ
- non_keywords 接続プロパティ733 ページ
- 「reserved_keywords オプション」673 ページ


## odbc_describe_binary_as_varbinary

SQL Anywhere ODBC ドライバーの BINARY カラムの記述方法を制御します。

### 指定可能な値

On、Off

### デフォルト

Off

### 備考

このオプションを使用すると、すべての BINARY カラムと VARBINARY カラムを、BINARY としてまたは VARBINARY として、アプリケーションに対して記述するかどうかを選択できます。デフォルトでは、SQL Anywhere ODBC ドライバーは BINARY と VARBINARY の両方のカラムを SQL_BINARY として記述します。このオプションを On に設定すると、ODBC ドライバーは BINARY と VARBINARY カラムを SQL_VARBINARY として記述します。このオプションの設定にかかわらず、BINARY カラムと VARBINARY カラムを区別することはできません。

BINARY カラムは常に 0 埋め込みで、VARBINARY カラムはそうではない Delphi アプリケーションを使用する場合は、このオプションを On に設定することをおすすめします。このオプションを On にしてすべてのカラムが可変長データ型として扱われるようにすると、Delphi のパフォーマンスが向上します。

### 参照

- 「BINARY データ型」『SQL Anywhere サーバー SQL リファレンス』
- 「VARBINARY データ型」『SQL Anywhere サーバー SQL リファレンス』
- odbc_describe_binary_as_varbinary 接続プロパティ733 ページ

## odbc_distinguish_char_and_varchar オプション

SQL Anywhere ODBC ドライバーの CHAR カラムの記述方法を制御します。

### 指定可能な値

On、Off

### デフォルト

Off

### 備考

接続を開くと、SQL Anywhere ODBC ドライバーは、このオプションの設定を使用して CHAR カラムの記述方法を決定します。このオプションを Off (デフォルト) に設定すると、CHAR カラムは SQL_VARCHAR として記述されます。このオプションを On に設定すると、CHAR カラムは SQL_CHAR として記述されます。VARCHAR カラムは、常に SQL_VARCHAR として記述されます。

### 参照

- 「CHAR データ型」『SQL Anywhere サーバー SQL リファレンス』
- 「VARCHAR データ型」『SQL Anywhere サーバー SQL リファレンス』
- odbc_distinguish_char_and_varchar 接続プロパティ733 ページ

# oem_string オプション

データベースファイルのヘッダーページ内にあるユーザー指定の情報を保存します。

### 指定可能な値

文字列 (最大 128 バイト)

### デフォルト

空の文字列

### スコープ

PUBLIC グループのみに設定できます。すぐに有効になります。DBA 権限が必要です。

### 備考

データベースファイルのヘッダーページ内の情報を保存しておき、アプリケーションから直接ファイルを読み込んで情報を抽出できます。このページはシステム DB 領域ファイルのヘッダーに保存されます。OEM 文字列に対して 128 バイトを超える値を指定すると、エラーが返されます。

スキーマバージョン、アプリケーション名、アプリケーションのバージョンなどの情報は、保存しておくと役に立つことがあります。また、アプリケーションがデータベースファイルを使用する前に検証のために読み込む文字列を保存することによって、アプリケーションはデータベースを起動しなくても OEM 文字列によってデータベースファイルがそのアプリケーションに関連付けられているかどうかを確認できます。つまり、アプリケーションを設計するときに、OEM 文字列を使用してデータベースファイルがそのアプリケーション用であることを検証するプロセスを組み込むことができます。さらに、ユーザーに表示するメタデータを抽出することもできます。

oem_string をシステム DB 領域ファイルのヘッダー内に設定するには、次の文を実行します。

```
SET OPTION PUBLIC.oem_string=user-specified-string;
```

*user-specified-string* の値は、ISYSOPTION システムテーブルとシステム DB 領域ファイルのヘッダーの両方に保存されます。この文字列が SET OPTION 文に渡されるときには変換が行われないため、文字列を必要な文字セットで定義してから SET OPTION 文に指定する必要があります。文字列を必要な文字セットに変換するには、CSCONVERT 関数を使用します。

oem_string の値を問い合わせるには、以下の方法があります。

●oem_string 接続プロパティを使用する。

```
SELECT CONNECTION_PROPERTY( 'oem_string' );
```

●SYSOPTION システムビューを使用する。

```
SELECT setting FROM SYSOPTION WHERE "option" = 'oem_string';
```

**♦ アプリケーションからの oem_string オプションの値の問い合わせ**

1. データベースシステム DB 領域ファイルを開きます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

2. このファイルの最初のページをバッファーに読み込みます。

3. バッファーから OEM 文字列の前後にある 2 バイトのプレフィクスとサフィックスを検索します。

   プレフィクスとサフィックスは、*sqldef.h* にそれぞれ DB_OEM_STRING_PREFIX と DB_OEM_STRING_SUFFIX として定義されています。これら 2 つの文字列に囲まれたすべてのバイトがデータベースに定義された OEM 文字列です。

SQL Anywhere には 2 つのサンプルプログラムが付属しており、*oem_string* ディレクトリに保存されています。

- *dboem.cpp* は、OEM 文字列を抽出してデータベースサーバーメッセージウィンドウに出力する方法を示す C プログラムです。

- *dboem.pl* は、OEM 文字列を抽出して PERL スクリプト内 の stdout に出力する方法を示します。

**警告**
アプリケーションからデータベース内の OEM 文字列に直接書き込みを行うことはできません。そのようにすると、データベースのヘッダーページが破損します。

Windows では、サーバーがこのデータベースファイルをロードした場合、アプリケーションからこのファイルを直接読み込むことはできません。データベースサーバーは、このファイルに排他ロックを設定します。これに対し、サポートされる UNIX プラットフォームでは、読み込みパーミッションのあるアプリケーションであれば、このファイルをいつでも直接読み込むことができます。ただし、OEM 文字列に対する変更は、直ちにファイルに反映されないことがあります。チェックポイントを発行すると、データベースサーバーがページ 0 をディスクにフラッシュするため、現在の OEM 文字列値がファイルに反映されます。

OEM 文字列を変更してから次のチェックポイントまでの間にデータベースサーバーで障害が発生すると、ファイルのヘッダーに新しい OEM 文字列値が反映されないことがあります。新しい OEM 文字列値は、データベースのリカバリが完了した後で正しく設定されます。

## 参照


- 「CSCONVERT 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』
- oem_string 接続プロパティ 733 ページ


## 例

次の例は、データベースファイルに関する情報を示す OEM 文字列を暗号化し、データベースヘッダーファイルに保存します。

```
BEGIN
  DECLARE @v VARCHAR(100);
  SET @v = BASE64_ENCODE( ENCRYPT( 'database version 10', 'abc' ) );
  EXECUTE IMMEDIATE 'SET OPTION PUBLIC.oem_string = '" || @v || "';
END;
```

OEM 文字列の値を取得するには、次の文を実行します。

```
SELECT DECRYPT(
  BASE64_DECODE(
    CONNECTION_PROPERTY( 'oem_string' ) ),'abc' )
```

## on_charset_conversion_failure オプション

文字の変換中にエラーが発生した場合の動作を制御します。

### 指定可能な値

Ignore、Warning、Error

### デフォルト

Ignore

### 備考

文字の変換中にエラーが発生した場合の動作を次のように制御します。

- **Ignore**　エラーも警告も表示しません。
- **Warning**　置換と不正な文字列を警告としてレポートします。不正な文字列は変換されません。
- **Error**　置換と不正な文字列をエラーとしてレポートします。

クライアントとデータベース間で文字セットの変換が必要な場合、不正な文字が検出されたり文字の置換が使用されたりしたときに、それを無視するか、警告を返すか、エラーを返すかをこのオプションで制御します。

シングルバイトからシングルバイトへの変換では、置換と不正な文字のレポートはできないので、Ignore に設定する必要があります。

このオプションは、損失を伴う変換がクライアントで発止した場合の動作は制御しません。たとえば、クライアントからの SQL 文は CHAR データベース文字セットに格納されているか、変換される必要があります。Unicode のクライアントアプリケーションが SQL 文を準備し、その文に CHAR データベース文字セットで表現できない文字が含まれているとします。この場合、置換文字が代わりに使用されます。ただし、損失を伴う変換がクライアントで発生したため、データベースサーバーではこの変換が発生したことがわかりません。

### 参照


- 「CHAR と NCHAR の比較」『SQL Anywhere サーバー SQL リファレンス』
- 「NCHAR から CHAR への変換」『SQL Anywhere サーバー SQL リファレンス』
- 「損失を伴う変換および置換文字」『SQL Anywhere サーバー SQL リファレンス』
- on_charset_conversion_failure 接続プロパティ 734 ページ



Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## on_tsql_error オプション

ストアドプロシージャーのエラー処理を制御します。

### 指定可能な値

- **Stop**　エラーの検出と同時に実行を停止します。
- **Conditional**　プロシージャーが ON EXCEPTION RESUME を使用していて、エラーのすぐ後ろの文がエラーを処理する場合は継続します。それ以外の場合は終了します。
- **Continue**　次に続く文に関係なく実行は継続されます。複数のエラーがある場合は、ストアドプロシージャーで最初に検出されたエラーが返されます。

### デフォルト

Conditional

jConnect 接続の場合は Continue

### 備考

このオプションは、ストアドプロシージャーと Transact-SQL バッチのエラー処理を制御します。

on_tsql_error の設定 Conditional と Continue は、いずれも Adaptive Server Enterprise との互換性を確保するために使用します。Continue が Adaptive Server Enterprise 動作を最も忠実にシミュレートします。エラーがより早期にレポートされるようにするには、Transact-SQL の新しいストアドプロシージャーを作成するときに Conditional 設定を使用します。

このオプションを Stop または Continue に設定した場合は、その設定が continue_after_raiserror オプションの設定より優先されます。ただし、このオプションを Conditional (デフォルト) に設定した場合は、RAISERROR 文の後の動作は continue_after_raiserror オプションによって決まります。

### 参照


- 「CREATE PROCEDURE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「CREATE PROCEDURE 文 [T-SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「Transact-SQL のプロシージャー言語」『SQL Anywhere サーバー SQL の使用法』
- 「continue_after_raiserror オプション」609 ページ
- 「互換性オプション」584 ページ
- 「sp_tsql_environment システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- on_tsql_error 接続プロパティ 734 ページ


## optimization_goal オプション

クエリ処理の最適化の対象を、最初のローを迅速に返すこと、または完全な結果セットを返すコストを最小限に抑えることのどちらかに指定します。

### 指定可能な値

First-row または All-rows

### デフォルト

All-rows

### 備考

optimization_goal オプションは、SQL Anywhere において SQL データ操作言語 (DML) 文を応答時間に対して最適化するか、リソースの総消費量に対して最適化するかを制御します。

このオプションを All-rows (デフォルト) に設定すると、SQL Anywhere はクエリを最適化して、予測される合計検索時間が最短になるアクセスプランを選択します。Sybase PowerBuilder DataWindow アプリケーションなど、処理の前に結果セット全体が必要になるアプリケーションでは、optimization_goal を All-rows に設定するのが適切です。All-rows の設定では、カーソルが開いたときに結果全体が実体化されるため、insensitive (ODBC の静的) カーソルにも適しています。また、結果セットのスクロールを目的とするスクロール (ODBC キーセット駆動型) カーソルにも適しています。

このオプションを First-row に設定すると、SQL Anywhere は、クエリの結果の最初のローをフェッチするまでの時間を短縮するアクセスプランを選択します。この場合、検索にかかる合計時間は長くなることがあります。また、通常 SQL Anywhere オプティマイザーでは、可能であれば結果の実体化を必要とするアクセスプランは使用しないで、最初のローを返すまでの時間を短縮します。この設定では、オプティマイザーは、明示的なソートの操作を必要とするアクセスプランではなく、クエリの ORDER BY 句を満たすインデックスを使用するアクセスプランを採用します。

クエリの FROM 句の FASTFIRSTROW テーブルヒントを使用すると、特定のクエリの最適化ゴールを First-row に設定できます。この場合、optimization_goal の設定を変更する必要はありません。

FASTFIRSTROW テーブルヒントの使用の詳細については、「FROM 句」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT、INTERSECT の各文に OPTION 句を含めることによって、各文で指定したオプション設定をこれらのオプションに対するテンポラリ設定やパブリック設定よりも優先させることができます。

### 参照

- 「INSERT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「UPDATE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「DELETE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「SELECT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「UNION 文」『SQL Anywhere サーバー SQL リファレンス』
- 「EXCEPT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「INTERSECT 文」『SQL Anywhere サーバー SQL リファレンス』
- optimization_goal 接続プロパティ 734 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

# optimization_level オプション

SQL Anywhere クエリオプティマイザーが SQL 文のアクセスプランの検索に費やす作業量を制御します。

### 指定可能な値

0-15

### デフォルト

9

### 備考

optimization_level オプションは、SQL Anywhere オプティマイザーが SQL データ操作言語 (DML) の最適化に費やす作業量を制御します。このオプションは、任意の SELECT ブロックについてオプティマイザーが考慮する代替のジョイン方式の最大数を制御します。optimization_level の設定値が高いほど、オプティマイザーが考慮するジョイン方式の最大数は大きくなります。

このオプションを 0 に設定すると、SQL Anywhere オプティマイザーは実行のために考慮する最初のアクセスプランを選択し、事実上、代替プランのコストベースの比較を避けることになります。さらに、レベル 0 では、ネストされたクエリのセマンティックな最適化が一部無効になります。このオプションが 0 よりも大きい値に設定されると、オプティマイザーは代替方式を評価し、予想コストが最も低いものを選択します。このオプションがデフォルトの 9 よりも大きい値に設定されると、オプティマイザーは代替方式をより積極的に検索し、その結果、最適化フェーズで経過する時間がはるかに長くなる可能性があります。

代表的なシナリオでは、アプリケーションが DML 文に対してより速い OPEN 時間を必要とする場合、このオプションは一時的に低いレベル (0、1、2 など) に設定されます。文が複雑であってもクエリの実行時間は非常に短いので、オプティマイザーによって選択された特定のアクセスプランはあまり重要ではないことがわかっています。optimization_level の PUBLIC 設定をデフォルトから変更することはおすすめできません。

optimization_level オプションの設定の結果は、optimization_goal と optimization_workload オプションの設定とは無関係です。

単純な DML 文 (特定の行を識別する WHERE 句に等号条件を含んだ単一ブロック、単一テーブルのクエリ) はヒューリスティックに最適化されるため、コストベースのオプティマイザーをすべてバイパスします。単純な DML 文の最適化は、optimization_level オプションの設定からは影響を受けません。オプティマイザーバイパスメカニズムによって最適化された要求の数は、QueryBypassed 接続プロパティとして使用できます。

QueryBypassed 接続プロパティの詳細については、「接続プロパティ値のアクセス」713 ページを参照してください。

INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT、INTERSECT の各文に OPTION 句を含めることによって、各文で指定したオプション設定をこれらのオプションに対するテンポラリ設定やパブリック設定よりも優先させることができます。

### 参照

- 「INSERT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「UPDATE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「DELETE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「SELECT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「UNION 文」『SQL Anywhere サーバー SQL リファレンス』
- 「EXCEPT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「INTERSECT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「オプティマイザーの仕組み」『SQL Anywhere サーバー SQL の使用法』
- optimization_level 接続プロパティ734 ページ

## optimization_workload オプション

クエリ処理において、更新と読み込みを組み合わせた負荷に対して最適化するか、または大部分が読み込みベースの負荷に対して最適化するかを決定します。

### 指定可能な値

Mixed、OLAP

### デフォルト

Mixed

### スコープ

PUBLIC グループのみに設定できます。DBA 権限が必要です。

### 備考

optimization_workload オプションは、SQL Anywhere が、更新と読み取りが組み合わさった負荷、または主に読み取りベースの負荷に対して、どちらのクエリ処理向けに最適化されるか制御します。

このオプションが Mixed (デフォルト) に設定されている場合、SQL Anywhere は短い挿入、更新、削除と、実行時間の長い読み取り専用クエリを組み合わせた負荷に対して適切なクエリ最適化アルゴリズムを選択します。

このオプションが OLAP に設定されている場合、SQL Anywhere は、実行時間の長いクエリの大部分とバッチ更新を組み合わせた負荷に対して適切なアルゴリズムを選択します。特に、オプティマイザーは、クラスタードハッシュ Group By クエリ実行アルゴリズムを使用するように選択する場合があります。

オプションが OLAP に設定されている場合、クラスタードハッシュ Group By アルゴリズムが有効になります。このオプションを Mixed (デフォルト) に設定すると、このアルゴリズムは無効になります。

INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT、INTERSECT の各文に OPTION 句を含めることによって、各文で指定したオプション設定をこれらのオプションに対するテンポラリ設定やパブリック設定よりも優先させることができます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照

- 「INSERT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「UPDATE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「DELETE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「SELECT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「UNION 文」『SQL Anywhere サーバー SQL リファレンス』
- 「EXCEPT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「INTERSECT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「optimization_level オプション」657 ページ
- optimization_workload 接続プロパティ734 ページ

## pinned_cursor_percent_of_cache オプション

カーソルを固定するために使用できるキャッシュの割合を指定します。

### 指定可能な値

整数 (0 ～ 100)

### デフォルト

10

### スコープ

PUBLIC グループのみに設定できます。DBA 権限が必要です。

### 備考

データベースサーバーは、カーソルの実装に必要なデータ構造を、仮想メモリのページに格納します。これらのページは、フェッチ要求から次のフェッチ要求までの間メモリ内にロックされているため、次のフェッチ要求を受信したときにすぐに使用できます。

メモリが少ない環境で、これらのページによって必要以上にキャッシュが占有されないように、カーソルの固定に使用するキャッシュの割合は制限されています。この制限を調整するには、pinned_cursor_percent_of_cache オプションを使用します。

このオプションの値は、0 ～ 100 の割合 (%) で指定します。デフォルト値は 10 です。0 に設定すると、次のフェッチ要求までの間カーソルページは固定されません。

### 参照

- pinned_cursor_percent_of_cache 接続プロパティ735 ページ

## post_login_procedure オプション

ユーザーが接続したときにアプリケーションが表示する必要のあるメッセージが結果セットに含まれるプロシージャーを指定します。

### 指定可能な値

文字列

### デフォルト

sa_post_login_procedure システムプロシージャー

### スコープ

DBA 権限が必要です。

### 備考

post_login_procedure を空の文字列以外に設定すると、アプリケーションは、このオプションに指定されたプロシージャーを接続プロセス中に呼び出し、ユーザーに対して表示するメッセージを特定できます。このオプションには、*owner.function-name* という形式の値を指定する必要があります。これによって、ユーザーが関数を無効にすることを防止できます。

このオプションを設定すると、Sybase Central 用の SQL Anywhere 12 プラグインである Interactive SQL がプロシージャーを呼び出し、プロシージャーが返すメッセージをウィンドウに表示します。SQL Anywhere に付属していないアプリケーションについては、必要に応じて、このオプションに指定されたプロシージャーを呼び出してメッセージを表示するように変更を加えてください。

接続時にアプリケーションがメッセージを表示することが必要となる状況としては、パスワード有効期限システムを実装している場合に、パスワードの期限切れが近づいていることをユーザーに通知することがあります。このような機能を利用すると、ユーザーが接続するたびに、パスワードが期限切れとなることを数日前に知らせることができます。

このオプションに指定するプロシージャーは、1 つ以上のローと 2 つのカラムで構成される結果セットを返す必要があります。1 つ目のカラムは VARCHAR(255) 型で、メッセージのテキスト (メッセージがない場合は NULL) を返します。2 つ目のカラムは INT 型で、アクションのタイプを返します。アクションとして指定可能な値は次のとおりです。

- **0**　メッセージを表示 (メッセージがある場合)
- **1**　メッセージを表示し、ユーザーにパスワードの変更を要求
- **2-99**　予約済み
- **100 以上**　ユーザー定義

SQL Anywhere 12 プラグインと Interactive SQL (dbisql) は、アクションの値にかかわらず、NULL 以外であれば、すべてのメッセージを表示します。アクションを 1 に設定すると、これらのツールはユーザーにパスワードを変更するよう要求し、新しいパスワードをユーザー指定の値に設定します。

post_login_procedure の使用と、パスワード有効期限の運用などの詳細なパスワード規則の実装を示す例については、「パスワード検証関数の使用」を参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照


- 「login_procedure オプション」636 ページ
- 「verify_password_function オプション」708 ページ
- 「パスワードのセキュリティ」1383 ページ
- 「NewPassword (NEWPWD) 接続パラメーター」355 ページ
- 「sa_post_login_procedure システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- post_login_procedure 接続プロパティ735 ページ


### 例

次の例では、p_post_login_check というプロシージャーを使用して、ユーザーにパスワードの期限切れが近づいていることを知らせ、パスワードを変更するよう要求します。

```
CREATE PROCEDURE DBA.p_post_login_check( )
RESULT( message_text VARCHAR(255), message_action INT )
BEGIN
  DECLARE message_text        CHAR(255);
  DECLARE message_action      INT;

  -- assume the password_about_to_expire variable was
  -- set by the login procedure
  IF password_about_to_expire = 1 THEN
    SET message_text = 'Your password is about to expire';
    SET message_action = 1;
  ELSE
    SET message_text = NULL;
    SET message_action = 0;
  END IF;
  -- return message (if any) through this result set
  SELECT message_text, message_action;
END;

GRANT EXECUTE ON DBA.p_post_login_check TO PUBLIC;

SET OPTION PUBLIC.post_login_procedure = 'DBA.p_post_login_check';
```

## precision オプション

10 進法計算での結果の最大桁数を指定します。

### 指定可能な値

整数 (1 ～ 127)

### デフォルト

30

### スコープ

PUBLIC グループのみに設定できます。すぐに有効になります。

### 備考

精度は、小数点の左右の合計桁数です。scale オプションは、計算結果が最大 precision にトランケートされた場合の、小数点以下の最小桁数を指定します。

掛け算、割り算、足し算、引き算、集合関数はすべて結果が最大精度を超える可能性があります。

たとえば、DECIMAL(8,2) と DECIMAL(9,2) を掛けると、結果には DECIMAL(17,4) が必要です。precision が 15 の場合、15 桁のみが結果に残ります。scale が 4 の場合、結果は DECIMAL(15.4) です。scale が 2 の場合、結果は DECIMAL(15,2) です。どちらの場合も、オーバーフローが発生する可能性があります。

### 参照


- 「scale オプション」676 ページ
- 「DECIMAL データ型」『SQL Anywhere サーバー SQL リファレンス』
- 「NUMERIC データ型」『SQL Anywhere サーバー SQL リファレンス』
- precision 接続プロパティ 735 ページ
- scale 接続プロパティ 743 ページ


## prefetch オプション

クライアントアプリケーションで使用できるようになる前に、ローがクライアント側にフェッチされるかどうかを制御します。

### 指定可能な値

Off、Conditional、Always

### デフォルト

Conditional

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

### 備考

このオプションは、クライアントアプリケーションで使用できるようになる前に、ローがクライアントサイドにフェッチされるかどうかを制御します。一度に複数のローをフェッチすると、クライアントアプリケーションが一度に 1 つずつローを要求した場合 (カーソルのローをループする場合など) でも、応答時間が短縮され、データベースへの要求数の減少によって全体的なスループットも向上します。

- Off はプリフェッチが行われないことを意味します。

- Conditional (デフォルト) の場合は、カーソルタイプが SENSITIVE か、クエリにプロキシテーブルが含まれないかぎり、プリフェッチが発生します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

●Always は、SENSITIVE タイプのカーソルの場合も、プロキシテーブルが関連するカーソルの場合も、プリフェッチが行われることを意味します。

Always は、一部のカーソルのセマンティックに影響するため、注意して使用してください。たとえば、SENSITIVE タイプのカーソルが ASENSITIVE タイプになる場合があります。また、プリフェッチと、アプリケーションのフェッチ要求の間に値が更新された場合は、古い値がフェッチされることがあります。さらに、プロキシテーブルが関連するカーソルでプリフェッチを使用した場合、クライアントがプリフェッチローを再フェッチしようとすると、エラー -668「カーソルは FETCH NEXT 操作に制限されています」が発生する可能性があります。最初のフェッチの後で、初めてフェッチカラムが再バインドまたはバインドされた場合、クライアントは、ロールバックの後、または 0 の相対フェッチ時、あるいは GET DATA が使用されるときに、プリフェッチローを再フェッチしようとすることがあります。

value-sensitive カーソルタイプには、ESQL SENSITIVE と SCROLL カーソルタイプ、ODBC と OLE DB DYNAMIC および KEYSET カーソルタイプが含まれます。

prefetch オプションの設定は、Sybase Open Client 接続と jConnect 接続では無視されます。

DisableMultiRowFetch 接続パラメーターが YES に設定されている場合、prefetch データベース オプションは無視され、プリフェッチは行われません。

このオプションでは、以前は On も有効な値でした。この値は現在は Conditional のエイリアスです。

### 参照


- 「プリフェッチ」『SQL Anywhere サーバー プログラミング』
- 「DisableMultiRowFetch (DMRF) 接続パラメーター」339 ページ
- prefetch 接続プロパティ 736 ページ


## preserve_source_format オプション

プロシージャー、トリガー、ビュー、イベントハンドラーの元のソース定義をシステムファイルに保存するかどうかを制御します。保存した場合は、SYSTAB、SYSPROCEDURE、SYSTRIGGER、SYSEVENT 内のカラム source に保存されます。

### 指定可能な値

On、Off

### デフォルト

On

### スコープ

PUBLIC グループのみに設定できます。DBA 権限が必要です。

### 備考

preserve_source_format を On に設定すると、データベースサーバーは、プロシージャー、ビュー、トリガー、イベントの CREATE 文と ALTER 文によってフォーマットされたソースを保存し、それを適切なシステムビューのソースカラムに配置します。

フォーマットされていないソーステキストは、同じシステムテーブルの proc_defn、trigger_defn、view_defn の各カラムに格納されます。ただし、これらの定義は、Sybase Central では簡単には読めません。フォーマットされたソースカラムでは、スペース、コメント、大文字または小文字を任意に選んで定義を参照できます。

このオプションを OFF にすると、データベースにオブジェクト定義を保存するために使用される領域を減らすことができます。このオプションは、PUBLIC ユーザーのみに設定できます。

### 参照

- preserve_source_format 接続プロパティ 736 ページ

## prevent_article_pkey_update オプション

Mobile Link パブリケーションに関連するテーブルのプライマリキーカラムの更新を制御します。

### 指定可能な値

On、Off

### デフォルト

On

### 備考

このオプションを On に設定すると、パブリケーションに含まれるテーブルのプライマリキーカラムの更新が禁止されます。このオプションを使用することで、特にレプリケーションと同期の環境におけるデータの整合性が確保されます。

> **警告**
> 同期環境やレプリケーション環境では、このオプションを Off に設定しないでください。

### 参照

- prevent_article_pkey_update 接続プロパティ 736 ページ

## priority オプション

接続からの要求を実行する優先度レベルを設定します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 指定可能な値

Critical、High、Above Normal、Normal、Below Normal、Low、Background

### デフォルト

Normal

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

このオプションを一時的に設定した場合、この設定は現在の接続にのみ適用されます。同じユーザー ID による別の接続では、このオプションの設定を変えることができます。

### 備考

このオプションには、max_priority オプションの値より高い値は設定できません。

### 参照


- 「max_priority オプション」643 ページ
- 「BACKUP 文」『SQL Anywhere サーバー SQL リファレンス』
- 「LOAD TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「REORGANIZE TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「RESTORE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「UNLOAD 文」『SQL Anywhere サーバー SQL リファレンス』
- 「VALIDATE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「接続リソースの使用」552 ページ
- priority 接続プロパティ736 ページ


## progress_messages オプション

データベースサーバーからクライアントに進行メッセージを送信するかどうかを制御します。

### 指定可能な値

- **Off**　クライアントに進行メッセージを送信しません。
- **Raw**　Raw を選択すると、進行メッセージに次のフォーマットが使用されます。

  43;9728;22230;pages;5025;6138

  Raw 進行メッセージには、セミコロンで区切られた 6 つのフィールドがあります。各フィールドの定義は、次のとおりです。

  - **フィールド 1**　実行済みの文の割合。
  - **フィールド 2**　完了したページ数、ロー数、またはバイト数。
  - **フィールド 3**　処理予定のページ数、ロー数、またはバイト数。
  - **フィールド 4**　処理中の内容：ページ、ロー、またはバイト。

○ **フィールド 5**　現在の経過時間 (ミリ秒で表示)。

○ **フィールド 6**　文の実行が完了するまでの残りの推定所要時間 (ミリ秒)。

● **Formatted**　Formatted を選択すると、進行メッセージに次のフォーマットが使用されます。

43% (22230 ページの 9728) が 00:00:05 後に完了します。残り約 00:00:06 です。

Formatted 進行メッセージはローカライズされ、時刻フォーマットは HH:MM:SS です。100 KB 未満の単位はバイト単位、100 MB 未満の単位はキロバイト (KB) 単位、100 MB 以上の単位はメガバイト (MB) 単位でそれぞれ表示されます。

## デフォルト

Off

## 備考

進行メッセージは、文の合計推定継続時間の 5% の間隔で送信されます。一般に、推定が完了すると、最初の進行メッセージが 10 秒以内に送信されます。以降の進行メッセージは 30 秒から 5 分の間隔で送信されます。完了した割合が前のメッセージで送信された値と同じ場合は、メッセージが最後に送信されてから 5 分以上経過するまで、更新された進行メッセージは送信されません。文の実行所要時間が 30 秒未満の場合には、進行メッセージは送信されません。

推定時間は継続的に再計算され、操作の進行に伴って推定残り時間の精度が高まります。バックアップなどのイベント時には、文の実行中に合計ページ数が調整されるため、完了した割合と推定残り時間が変わります。BACKUP...WITH CHECKPOINT COPY または UNLOAD SELECT などの文では、影響を受けるページまたはローの合計数が不明なため、完了した割合の値が 100% を超える可能性があります。その結果、推定残り時間は計算できず、進行メッセージには含まれません。

次の文とプロシージャーでは、進行メッセージがサポートされています。

- BACKUP DATABASE (イメージとアーカイブの両方)
- LOAD TABLE (USING FILE と USING CLIENT FILE のみ)
- MESSAGE
- REORGANIZE TABLE
- RESTORE DATABASE
- UNLOAD (すべてのタイプ)
- sa_table_page_usage システムプロシージャー

ユーティリティデータベースに接続されている場合は、SET TEMPORARY OPTION 文を使用して progress_messages オプションを設定できます。「SET OPTION 文」『SQL Anywhere サーバー SQL リファレンス』と「ユーティリティデータベースに使用できる文」136 ページを参照してください。

また、**[ツール]** » **[オプション]** » **[SQL Anywhere]** » **[コマンド]** » **[進行メッセージを表示]** をクリックして、Interactive SQL で progress_messages オプションを設定することもできます。**[進行メッセージを表示]** を有効にすると、progress_messages オプションが Formatted に設定されます。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照

- 「BACKUP 文」『SQL Anywhere サーバー SQL リファレンス』
- 「バックアップユーティリティ (dbbackup)」913 ページ
- 「LOAD TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「MESSAGE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「REORGANIZE TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「RESTORE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「UNLOAD 文」『SQL Anywhere サーバー SQL リファレンス』
- 「VALIDATE 文」『SQL Anywhere サーバー SQL リファレンス』
- Progress 接続プロパティ 736 ページ
- 「sa_table_page_usage システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』

## qualify_owners オプション [SQL Remote]

SQL Remote によってレプリケートされる SQL 文で、修飾されたオブジェクト名を使用するかどうかを指定します。

### 指定可能な値

On、Off

### デフォルト

On

### 備考

SQL Anywhere のインストール環境で修飾が必要でない場合は、オプションを Off にすると、メッセージが少し小さくなります。

### 参照

- 「SQL Remote オプション」『SQL Remote』

## query_mem_timeout オプション

要求にメモリが付与されるまでの最大待機時間をミリ秒で設定します。

### 指定可能な値

-1、0、正の整数

### デフォルト

-1

### 備考

このオプションを -1 (デフォルト) または 0 より小さい値に設定すると、要求にメモリが付与されるまでの待機時間は、要求の推定実行時間より最大で 50 倍の長さになります。このオプショ

ンを 0 に設定すると、要求はメモリが付与されるまで永久に待機します。それ以外の場合、設定したミリ秒の値が要求にメモリが付与されるまでの最大待機時間になります。

### 参照

- 「キャッシュとメモリガバナー」『SQL Anywhere サーバー SQL の使用法』
- query_mem_timeout 接続プロパティ 737 ページ

## quote_all_identifiers オプション [SQL Remote]

SQL Remote によってレプリケートされる SQL 文で、引用符で囲んだ識別子を使用するかどうかを指定します。

### 指定可能な値

On、Off

### デフォルト

Off

### 備考

このオプションを Off に設定すると、dbremote が SQL Anywhere で引用符が必要な識別子を (通常行われているように) 引用符で囲みます。

このオプションが On の場合、すべての識別子が引用符で囲まれます。

### 参照

- 「SQL Remote オプション」『SQL Remote』

## quoted_identifier オプション

二重引用符内の文字列の解釈を制御します。

### 指定可能な値

On、Off

### デフォルト

On

Sybase Open Client 接続と jConnect 接続の場合は Off

### 備考

このオプションは、二重引用符内の文字列を、識別子 (On) とリテラル文字列 (Off) のどちらとして解釈するかを制御します。quoted_identifier オプションは、Transact-SQL との互換性を保つために実装されています。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

「Transact-SQL 互換性のオプション」『SQL Anywhere サーバー SQL の使用法』を参照してください。

### 参照

- 「互換性オプション」584 ページ
- 「sp_tsql_environment システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「SET 文 [T-SQL]」『SQL Anywhere サーバー SQL リファレンス』
- quoted_identifier 接続プロパティ738 ページ

## read_past_deleted オプション

独立性レベル 1 または 2 でコミットされていない削除におけるデータベースサーバー動作を制御します。

### 指定可能な値

On、Off

### デフォルト

On

### 備考

read_past_deleted が On (デフォルト) の場合、独立性レベル 1 または 2 の逐次スキャンではコミットされていない削除ローを省略します。Off の場合、(削除トランザクションがコミットまたはロールバックするまで) 逐次スキャンは独立性レベル 1 または 2 のコミットされていない削除ローをブロックします。このオプションは、独立性レベル 1 と 2 のサーバー動作を変更します。

ほとんどの場合、このオプションは On のままにしてください。Off に設定すると、(使用可能なインデックスがある場合) ブロック動作はオプティマイザーが選択したプランに左右されます。

### 参照

- 「独立性レベルと一貫性」『SQL Anywhere サーバー SQL の使用法』
- read_past_deleted 接続プロパティ739 ページ

## recovery_time オプション

データベースサーバーがシステム障害から回復するのにかかる最長時間を分で設定します。

### 指定可能な値

整数 (分)

### デフォルト

2

### スコープ

PUBLIC グループのみに設定できます。DBA 権限が必要です。サーバー再起動時に有効になります。

### 備考

このオプションは、いつチェックポイントを実行すべきかを決定するために、checkpoint_time オプションと一緒に使用します。

SQL Anywhere はヒューリスティックを使用して、最後のチェックポイント以後に行った操作に基づいてリカバリ時間を予測します。この予測には、データベースのリカバリ予想時間とチェックポイント予想時間も含まれます。そのため、リカバリ時間は正確ではありません。

### 参照


- 「自動リカバリ処理」1089 ページ
- 「checkpoint_time オプション」603 ページ
- 「-gr dbeng12/dbsrv12 サーバーオプション」242 ページ
- 「データベースサーバーがチェックポイントのタイミングを決定する方法」1114 ページ
- recovery_time 接続プロパティ739 ページ


## remote_idle_timeout オプション

Web サービスのクライアントプロシージャーとファンクションで許容される休止時間 (秒数) を制御します。

### 指定可能な値

整数 (秒)

### デフォルト

15

### 備考

このオプションは、Web サービスのクライアントプロシージャーとファンクションに影響します。アクティビティのない状態が指定の秒数を越えると、プロシージャーまたはファンクションはタイムアウトになります。

### 参照


- 「Web クライアントアプリケーションの開発」『SQL Anywhere サーバー プログラミング』
- remote_idle_timeout 接続プロパティ739 ページ
- 「CREATE PROCEDURE 文 [Web サービス]」『SQL Anywhere サーバー SQL リファレンス』
- 「CREATE FUNCTION 文 [Web サービス]」『SQL Anywhere サーバー SQL リファレンス』


 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## replication_error オプション [SQL Remote]

SQL エラーが発生したときに Message Agent で呼び出すストアドプロシージャーを指定できます。

### 指定可能な値

ストアドプロシージャー名

### デフォルト

プロシージャーなし

### 備考

SQL Remote で replication_error オプションを使用すると、SQL エラーが生じたときに Message Agent で呼び出すストアドプロシージャーを指定できるようにします。デフォルトではプロシージャーを呼び出しません。

プロシージャーには、データ型が CHAR、VARCHAR、または LONG VARCHAR の引数を 1 つ指定してください。プロシージャーは、SQL エラーメッセージで 1 回、エラーの原因となった SQL 文でもう 1 回呼び出されます。状況によっては (外部キー違反の場合など)、エラーの原因となった SQL 文が使用できないために、ストアドプロシージャーを 1 回しか呼び出すことができない場合があります。

このオプションを使用すると、レプリケーションで SQL エラーの追跡とモニターができますが、その場合でもやはり、エラーが起こらないように設計することが必要です。このオプションは、そのようなエラーを解決するためのものではありません。

### 参照


- 「SQL Remote オプション」『SQL Remote』
- 「replication_error_piece オプション [SQL Remote]」671 ページ
- 「エラー処理プロシージャーの実行」『SQL Remote』


## replication_error_piece オプション [SQL Remote]

replication_error オプションとともに使用することにより、SQL Remote レプリケーション中に SQL エラーが発生した場合に Message Agent によって呼び出される LONG VARCHAR ストアドプロシージャーを指定できます。

### 指定可能な値

ストアドプロシージャー名

### デフォルト

プロシージャーなし

### 備考

エラーが発生し、replication_error が定義されている場合は、完全なエラー文字列とともに replication_error プロシージャーが呼び出されます。

replication_error と replication_error_piece の両方が定義されている場合、エラーは VARCHAR 部に分割されます。replication_error が最初の VARCHAR 部とともに呼び出され、replication_error_piece が残りの VARCHAR 部とともに繰り返し呼び出されます。

### 参照


- 「replication_error オプション [SQL Remote]」671 ページ
- 「SQL Remote オプション」『SQL Remote』


## request_timeout オプション

1 つの要求を実行できる最大時間を設定します。このオプションを使用すると、接続が大量のサーバーリソースを長時間にわたり消費することを防止できます。

### 指定可能な値

整数 (秒)、範囲は 0 ～ 86400 (1 日)

### デフォルト

0

### 備考

このオプションを 0 に設定すると、要求はタイムアウトになりません。

要求の実行時間が request_timeout に設定された時間 (CPU 時間ではなく実際の時間) を超えた場合、その要求は中断され、ユーザーにエラーが返されます。返されるエラーは、SQLE_REQUEST_TIMEOUT「タイムアウトになったため、要求が中断されました」です。要求がブロックされた場合、blocking_timeout オプションが 0 に設定されていると、要求は最大で request_timeout に指定された秒数の間ブロックされたままとなり、その時間の経過後、ブロッキングエラー (SQLE_LOCKED「'%2' のローは、ユーザー '%1' によってロックされています」など) を返します。

1 ～ 14 の範囲の値は、ユーザーオプションまたは PUBLIC オプションとしては指定できませんが、テンポラリオプションとしては指定できます。これによって、接続に時間がかかる場合でも (ログインプロシージャーが複雑であるなどの理由で)、ユーザーがデータベースサーバーに接続できなくなることはありません。

このオプションは、データベースクライアント要求と HTTP/HTTPS 要求のいずれに対しても使用できます。ただし、このオプションをストアドプロシージャーや HTTP/HTTPS 要求に設定しても、現在の要求には影響しません。要求の先頭にあるオプション値が使用されるからです。

request_timeout パブリックオプションを設定する場合は注意が必要です。このオプションを設定すると、実行時間の長い要求を発行するアプリケーション (dbvalid、dbbackup、dbunload など) が失敗する可能性があります。また、大量のサーバーリソースは使用しなくても別のユーザーで

ブロックされる可能性のあるアプリケーションも、request_timeout が設定されていると失敗することがあります。このような問題に対処するには、ログインプロシージャーでは接続の APPINFO 値に基づいて特定のアプリケーションだけに request_timeout オプションを設定します。

多数のローを含む結果セットをフェッチするなど、個々の要求を高速で評価するアプリケーションでは、このオプションを設定しても大量のサーバーリソースの消費を回避できないことがあります。

### 参照


- 「blocking_timeout オプション」602 ページ
- 「AppInfo (APP) 接続パラメーター」321 ページ
- request_timeout 接続プロパティ 742 ページ


## reserved_keywords オプション

デフォルトで無効になっている個々のキーワードをオンにします。

### 指定可能な値

文字列

### デフォルト

空の文字列

### 備考

このオプションは PUBLIC グループのみに設定できます。

ユーザーとテンポラリの設定は許可されていません。

このオプションは、デフォルトで無効になっている個々のキーワードをオンにします。LIMIT キーワードのみをオンにできます。

次の文では、LIMIT キーワードをキーワードとして認識できます。

```
SET OPTION PUBLIC reserved_keywords = 'LIMIT';
```

キーワードセット、単一オプション、複数オプションをオンにすることはできません。単語がキーワードとして識別されるかどうかは、次の条件に従って判別されます (優先度順)。

- その単語が予約語の SQL Anywhere リストに表示される。
- その単語が reserved_keywords オプションでオンになっている。
- その単語が non_keywords オプションでオフになっている。

このオプションで新規設定を行うと、前に設定しているものに置き換わります。次の文は以前の設定内容をすべてクリアします。

```
SET OPTION reserved_keywords =;
```

non_keywords オプションを使用すると、個々のキーワードをオフにできます。

### 参照

- 「non_keywords オプション」650 ページ
- 「キーワード」『SQL Anywhere サーバー SQL リファレンス』
- 「予約語」『SQL Anywhere サーバー SQL リファレンス』

## return_date_time_as_string オプション

クエリが実行されたときに、DATE、TIME、または TIMESTAMP の値がクライアントアプリケーションに渡される方法を制御します。

### 指定可能な値

On、Off

### デフォルト

Off

### スコープ

現在の接続の間、テンポラリオプションとしてのみ設定できます。

### 備考

このオプションは、DATE、TIME、または TIMESTAMP の値をアプリケーションに返すときに、DATE、TIME、または TIMESTAMP のデータ型とするか文字列とするかを指定します。

このオプションを On に設定すると、date_format、time_format、または timestamp_format オプションの設定を保持するために、データベースサーバーは DATE、TIME、または TIMESTAMP の値を文字列に変換してからクライアントに送信します。

Sybase Central と Interactive SQL では、return_date_time_as_string オプションは自動的に On に設定されます。

### 参照

- 「date_format オプション」613 ページ
- 「time_format オプション」696 ページ
- 「timestamp_format オプション」698 ページ
- 「timestamp_with_time_zone_format オプション」700 ページ
- return_date_time_as_string 接続プロパティ742 ページ

## rollback_on_deadlock

デッドロック発生時のトランザクションの処理方法を制御します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 指定可能な値

On、Off

### デフォルト

On

### スコープ

任意のユーザーによって設定可能で、PUBLIC グループと個々の接続に設定できます。すぐに有効になります。

### 備考

このオプションを On に設定すると、デッドロックが発生した場合、トランザクションは自動的にロールバックされます。ロールバックは、現在の要求が完了した後で発生します。このオプションを Off に設定した場合、SQL Anywhere はデッドロックが発生した文を自動的にロールバックし、そのトランザクションに対して、発生したデッドロックの種類を示すエラーメッセージを返します。文のロールバックでは、その文によって取得されたロックが解放される可能性はあまりありません。

### 参照


- 「デッドロック」『SQL Anywhere サーバー SQL の使用法』
- rollback_on_deadlock 接続プロパティ 743 ページ


## row_counts オプション

クエリを開くときに、データベースがローの数をカウントするかどうかを指定します。

### 指定可能な値

On、Off

### デフォルト

Off

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

### 備考

このオプションを Off に設定すると、通常、ローの数は推定値のみとなります。On に設定すると、ローの正確な数がカウントされます。

> **警告**
> row_counts を On に設定すると、クエリの実行時間がかなり長くなることがあります。On に設定した場合、SQL Anywhere はクエリを 2 回実行するので、実行時間が倍になります。

### 参照

- C API：sqlany_num_rows メソッド [SQL Anywhere C]『SQL Anywhere サーバー プログラミング』 sqlany_num_rows メソッド [SQL Anywhere C]『SQL Anywhere サーバー プログラミング』
- PHP API：「sasql_num_rows」『SQL Anywhere サーバー プログラミング』 「sasql_num_rows」『SQL Anywhere サーバー プログラミング』
- row_counts 接続プロパティ 743 ページ

## save_remote_passwords オプション [SQL Remote]

メッセージリンクに入力されたパスワードを保存します。

### 指定可能な値

On、Off

### デフォルト

On

### 備考

メッセージリンクパラメーターをデータベースではなく外部に保存する場合、パスワードが保存されないように設定できます。このオプションを Off に設定すると、パスワードは保存されません。

### 参照

- 「SQL Remote オプション」『SQL Remote』

## scale オプション

計算結果が最大 precision にトランケートされる場合の、小数点以下の最小桁数を指定します。

### 指定可能な値

precision データベースオプションに指定した値より小さい 0 ～ 127 の整数

### デフォルト

6

### スコープ

PUBLIC グループのみに設定できます。すぐに有効になります。

### 備考

掛け算、割り算、足し算、引き算、集合関数はすべて結果が最大精度を超える可能性があります。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照

- 「precision オプション」661 ページ
- scale 接続プロパティ743 ページ
- precision 接続プロパティ735 ページ

## secure_feature_key

データベースサーバーの -sf オプションによって保護された機能を接続において有効にできます。

### 指定可能な値

文字列

### デフォルト

NULL

### スコープ

現在の接続の間、テンポラリオプションとしてのみ設定できます。

### 備考

データベースサーバーを起動するときに -sf オプションを指定することにより、サーバー上で動作するデータベースが使用できない機能を指定できます。-sk サーバーオプションを使用すると、接続に対して保護されている (無効になっている) すべての機能を再び有効にするためのキーを指定できます。また、データベースサーバー上で実行されているすべてのデータベースに対して保護する機能を変更するための接続権限を付与します。データベースサーバーの起動時に、secure_feature_key テンポラリオプションの値を -sk オプションで指定されている値に設定すると、そのデータベース接続に対するすべての機能が再び有効化され、その接続では sa_server_option システムプロシージャーを使用してデータベース機能へのアクセスを制御できます。

secure_feature_key オプションを -sk で指定された値以外の値に設定すると、エラーは発生せず、-sf で指定された機能はその接続において無効なままとなります。

### 参照

- 「-sk dbeng12/dbsrv12 サーバーオプション」275 ページ
- 「-sf dbeng12/dbsrv12 サーバーオプション」271 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「機能の保護」1387 ページ
- secure_feature_key 接続プロパティ743 ページ

### 例

次のコマンドは、要求ログへのアクセスとすべてのリモートデータアクセス機能を無効にしてデータベースサーバー secure_server を起動します。これらの機能は、-sk で指定したキーを使用して、特定のデータベース接続において有効にできます。

```
dbsrv12 -n secure_server -sf request_log,remote -sk j978kls12 testdb.db
```

secure_server データベースサーバー上のデータベースに接続するときに secure_feature_key オプションを -sk で指定された値に設定すると、その接続において、要求ログへのアクセスとリモートデータアクセス機能が有効になります。

```
SET TEMPORARY OPTION secure_feature_key = 'j978kls12';
```

## sort_collation オプション

ORDER BY 式に対して SORTKEY 関数の暗黙の使用を許可します。

### 指定可能な値

Internal、collation_name、collation_id

### デフォルト

Internal

### 備考

このオプションの値が Internal である場合、ORDER BY 句は変わりません。

このオプションの値を、有効な照合名または照合 ID に設定すると、ORDER BY 句の CHAR または NCHAR 文字列式は、SORTKEY 関数が呼び出されたものとして扱われます。BINARY、UUID、XML、VARBIT などのその他の文字列データ型を使用する文字列式は、変更されません。

### 参照

- 「SORTKEY 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』
- sort_collation 接続プロパティ744 ページ

### 例

ソート照合をバイナリに設定します。

```
SET TEMPORARY OPTION sort_collation='binary';
```

ソート照合をバイナリに設定すると、以下のクエリに対して次のような変換が行われます。

```
SELECT Name, ID
FROM Products
ORDER BY Name, ID;
SELECT name, ID
FROM Products
ORDER BY 1, 2;
```

クエリは以下のように変換されます。

```
SELECT Name, ID
FROM Products
ORDER BY SORTKEY(Name, 'binary'), ID;
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

# sql_flagger_error_level オプション

指定した規格にはない SQL への応答を制御します。

### 指定可能な値

- Off
- SQL:1992/Entry
- SQL:1992/Intermediate
- SQL:1992/Full
- SQL:1999/Core
- SQL:1999/Package
- SQL:2003/Core
- SQL:2003/Package
- SQL:2008/Core
- SQL:2008/Package
- UltraLite

### デフォルト

Off

### 備考

このオプションは、指定した規格にはない SQL をエラーとして通知します。たとえば、SQL:2003/Package を指定すると、データベースサーバーは、上級レベルの SQL/2003 構文ではない構文として通知します。

デフォルトの動作 (Off) では、エラー通知はオフに設定されます。

SQL Anywhere の以前のバージョンとの互換性を持たせるため、次の値も受け入れられます。

- **E** このオプションは、SQL:1992/Entry に対応します。
- **I** このオプションは、SQL:1992/Intermediate に対応します。
- **F** このオプションは、SQL:1992/Full に対応します。
- **W** このオプションは、Off に対応します。

### 参照

- 「sa_ansi_standard_packages システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「SQLFLAGGER 関数 [その他]」『SQL Anywhere サーバー SQL リファレンス』
- 「sql_flagger_warning_level オプション」680 ページ
- 「SQL プリプロセッサー」『SQL Anywhere サーバー プログラミング』
- 「互換性オプション」584 ページ
- sql_flagger_error_level 接続プロパティ744 ページ

# sql_flagger_warning_level オプション

指定した規格にはない SQL への応答を制御します。

### 指定可能な値

- Off
- SQL:1992/Entry
- SQL:1992/Intermediate
- SQL:1992/Full
- SQL:1999/Core
- SQL:1999/Package
- SQL:2003/Core
- SQL:2003/Package
- SQL:2008/Core
- SQL:2008/Package
- UltraLite

### デフォルト

Off

### 備考

このオプションは、指定した規格にはない SQL を警告として通知します。たとえば、SQL:2003/Package を指定すると、データベースサーバーは、上級レベルの SQL/2003 構文ではない構文として通知します。

デフォルトの動作 (Off) では、警告通知はオフに設定されます。

以前のバージョンとの互換性を持たせるため、次の値も受け入れられます。

- **E** このオプションは、SQL:1992/Entry に対応します。
- **I** このオプションは、SQL:1992/Intermediate に対応します。
- **F** このオプションは、SQL:1992/Full に対応します。
- **W** このオプションは、Off に対応します。

### 参照


- 「sa_ansi_standard_packages システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「SQLFLAGGER 関数 [その他]」『SQL Anywhere サーバー SQL リファレンス』
- 「sql_flagger_error_level オプション」679 ページ
- 「SQL プリプロセッサー」『SQL Anywhere サーバー プログラミング』
- 「互換性オプション」584 ページ
- sql_flagger_warning_level 接続プロパティ745 ページ


 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## sr_date_format オプション [SQL Remote]

データベースから取り出した DATE 値のフォーマットを設定します。

### 指定可能な値

文字列 (次の記号の組み合わせ)

### デフォルト

yyyy-mm-dd

### 備考

日付を保存するカラムをレプリケートするときに Message Agent が使用します。フォーマットは次の記号を組み合わせた文字列です。

| 記号 | 説明 |
|---|---|
| YY | 2 桁の年 |
| YYYY | 4 桁の年 |
| MM | 2 桁の月 |
| MMM[m...] | 月を略した文字、"m" の数だけ文字を使用 |
| DD | 2 桁の日 |

各記号は、フォーマットされる日付のデータで置き換えられます。

文字データを表す記号 (*MMM* など) では、出力の文字を次のように制御できます。

- 記号をすべて大文字で入力すると、フォーマットはすべて大文字で表記されます。たとえば、MMM と入力すると、JAN と表記されます。
- 記号をすべて小文字で入力すると、フォーマットはすべて小文字で表記されます。たとえば、mmm と入力すると、jan と表記されます。
- 大文字と小文字を混ぜて入力すると、使用される言語に適切な文字が SQL Anywhere により選択されます。たとえば、Mmm と入力すると、英語では May、フランス語では mai と表記されます。

文字データがマルチバイトである場合、各記号の長さはバイト数ではなく文字数を表します。たとえば 'mmm' は 3 文字の月名を示しています。

数値データを表す記号では、記号に大文字を使用するか小文字を使用するかで 0 埋め込みを制御できます。

このオプションは、次の記号で構成される文字列です。

●記号をすべて大文字または小文字 (MM や mm など) で入力すると、0 埋め込みが行われます。たとえば、yyyy/mm/dd と入力すると、2002/01/01 と表記されます。

●大文字と小文字を混ぜて入力すると (Mm など)、0 の埋め込みは行われません。たとえば、yyyy/Mm/Dd と入力すると、2002/1/1 と表記されます。

### 参照


- 「sr_time_format オプション [SQL Remote]」682 ページ
- 「sr_timestamp_format [SQL Remote]」683 ページ
- 「sr_timestamp_with_time_zone_format [SQL Remote]」684 ページ
- 「SQL Remote オプション」『SQL Remote』


## sr_time_format オプション [SQL Remote]

データベースから取り出した TIME 値のフォーマットを設定します。

### 指定可能な値

文字列 (次の記号の組み合わせ)

### デフォルト

hh:nn:ss.Ssssss

### 備考

時間を保存するカラムをレプリケートするときに Message Agent が使用します。フォーマットは次の記号を組み合わせた文字列です。

| 記号 | 説明 |
|---|---|
| HH | 2 桁の時間 (24 時間表示)。 |
| NN | 2 桁の分。 |
| MM | コロンの後の場合は、2 桁の分 (HH:MM など)。 |
| SS.SSSSSS | 秒と小数点第 6 位までの秒の小数。小数点以下 6 桁のタイムスタンプをサポートしないプラットフォームもあります。 |

各記号は、フォーマットしようとする日付のデータで置き換えられます。

数値データを表す記号では、記号に大文字を使用するか小文字を使用するかで 0 埋め込みを制御できます。

●記号をすべて大文字または小文字 (NN や nn など) で入力すると、0 埋め込みが行われます。たとえば hh:nn:ss.sss と入力すると、02:34:56.123 と表記されます。

●大文字と小文字を混ぜて入力すると (Nn など)、0 の埋め込みは行われません。たとえば Hh:nn:ss.sss と入力すると、2:34:56.123 と表記されます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

- 小数点以下の秒の最初の 2 桁の大文字と小文字を混ぜて入力すると (Ss または sSssss)、後ろの 0 が削除されます。たとえば hh:nn:ss.Sss と入力すると、12:34:56.1 と表記されます。

### 備考

フォーマット文字列で大文字と小文字を混在させて使用すると、数字の前にゼロが付きません。

### 参照


- 「sr_date_format オプション [SQL Remote]」681 ページ
- 「sr_timestamp_format [SQL Remote]」683 ページ
- 「sr_timestamp_with_time_zone_format [SQL Remote]」684 ページ
- 「SQL Remote オプション」『SQL Remote』


## sr_timestamp_format [SQL Remote]

データベースから取り出した TIMESTAMP 値のフォーマットを設定します。

### デフォルト

yyyy-mm-dd hh:nn:ss.Ssssss

### 備考

このオプションは、Message Agent で DATETIME と TIMESTAMP の情報をレプリケートするときに使用します。

フォーマットは次の記号を組み合わせた文字列です。

| 記号 | 説明 |
|---|---|
| YY | 2 桁の年 |
| YYYY | 4 桁の年 |
| MM | 2 桁の月、または 2 桁の分 (HH:MM のように、コロンの後ろに続く場合) |
| MMM[m...] | 月を略した文字、"m" の数だけ文字を使用 |
| DD | 2 桁の日 |
| DDD[d...] | 曜日を略した文字 |
| HH | 2 桁の時間 |
| NN | 2 桁の分 |
| SS.SSSSSS | 秒と小数点第 6 位までの秒の小数。小数点以下 6 桁のタイムスタンプをサポートしないプラットフォームもあります。 |

| 記号 | 説明 |
|---|---|
| AA | A.M. または P.M. (12 時間表記) - 24 時間表記の場合、AA と PP は省略します。 |
| PP | P.M. (必要に応じて) (12 時間表記)−24 時間表記の場合、AA と PP は省略します。 |

各記号は、フォーマットされる日付のデータで置き換えられます。

文字データを表す記号 (*MMM* など) では、出力の文字を次のように制御できます。

- 記号をすべて大文字で入力すると、フォーマットはすべて大文字で表記されます。たとえば、MMM と入力すると、JAN と表記されます。
- 記号をすべて小文字で入力すると、フォーマットはすべて小文字で表記されます。たとえば、mmm と入力すると、jan と表記されます。
- 大文字と小文字を混ぜて入力すると、使用される言語に適切な文字が SQL Anywhere により選択されます。たとえば、Mmm と入力すると、英語では May、フランス語では mai と表記されます。

文字データがマルチバイトである場合、各記号の長さはバイト数ではなく文字数を表します。たとえば 'mmm' は 3 文字の月名を示しています。

数値データを表す記号では、記号に大文字を使用するか小文字を使用するかで 0 埋め込みを制御できます。

- 記号をすべて大文字または小文字 (MM や mm など) で入力すると、0 埋め込みが行われます。たとえば、yyyy/mm/dd と入力すると、2002/01/01 と表記されます。
- 大文字と小文字を混ぜて入力すると (Mm など)、0 の埋め込みは行われません。たとえば、yyyy/Mm/Dd と入力すると、2002/1/1 と表記されます。
- 小数点以下の秒の最初の 2 桁の大文字と小文字を混ぜて入力すると (Ss または sSssss)、後ろの 0 が削除されます。たとえば hh:nn:ss.Sss と入力すると、12:34:56.1 と表記されます。

**参照**


- 「sr_date_format オプション [SQL Remote]」 681 ページ
- 「sr_time_format オプション [SQL Remote]」 682 ページ
- 「sr_timestamp_with_time_zone_format [SQL Remote]」 684 ページ
- 「SQL Remote オプション」『SQL Remote』


## sr_timestamp_with_time_zone_format [SQL Remote]

データベースから取り出した TIMESTAMP WITH TIME ZONE 値のフォーマットを設定します。

### デフォルト

yyyy-mm-dd hh:nn:ss.Ssssss +hh:nn

### 備考

このオプションは、Message Agent で TIMESTAMP WITH TIME ZONE 情報をレプリケートするときに使用します。

フォーマットは次の記号を組み合わせた文字列です。

| 記号 | 説明 |
|---|---|
| YY | 2 桁の年 |
| YYYY | 4 桁の年 |
| MM | 2 桁の月、または 2 桁の分 (HH:MM のように、コロンの後ろに続く場合) |
| MMM[m...] | 月を略した文字、"m" の数だけ文字を使用 |
| DD | 2 桁の日 |
| DDD[d...] | 曜日を略した文字 |
| HH | 2 桁の時間 |
| NN | 2 桁の分 |
| SS.SSSSSS | 秒と小数点第 6 位までの秒の小数。小数点以下 6 桁のタイムスタンプをサポートしないプラットフォームもあります。 |
| AA | A.M. または P.M. (12 時間表記) − 24 時間表記の場合、AA と PP は省略します。 |
| PP | P.M. (必要に応じて) (12 時間表記) − 24 時間表記の場合、AA と PP は省略します。 |

各記号は、フォーマットされる日付のデータで置き換えられます。

文字データを表す記号 (*MMM* など) では、出力の文字を次のように制御できます。

- 記号をすべて大文字で入力すると、フォーマットはすべて大文字で表記されます。たとえば、MMM と入力すると、JAN と表記されます。
- 記号をすべて小文字で入力すると、フォーマットはすべて小文字で表記されます。たとえば、mmm と入力すると、jan と表記されます。
- 大文字と小文字を混ぜて入力すると、使用される言語に適切な文字が SQL Anywhere により選択されます。たとえば、Mmm と入力すると、英語では May、フランス語では mai と表記されます。

文字データがマルチバイトである場合、各記号の長さはバイト数ではなく文字数を表します。たとえば 'mmm' は 3 文字の月名を示しています。

数値データを表す記号では、記号に大文字を使用するか小文字を使用するかで 0 埋め込みを制御できます。

- 記号をすべて大文字または小文字 (MM や mm など) で入力すると、0 埋め込みが行われます。たとえば、yyyy/mm/dd と入力すると、2002/01/01 と表記されます。

- 大文字と小文字を混ぜて入力すると (Mm など)、0 の埋め込みは行われません。たとえば、yyyy/Mm/Dd と入力すると、2002/1/1 と表記されます。

- 小数点以下の秒の最初の 2 桁の大文字と小文字を混ぜて入力すると (Ss または sSssss)、後ろの 0 が削除されます。たとえば hh:nn:ss.Sss と入力すると、12:34:56.1 と表記されます。

### 参照


- 「sr_date_format オプション [SQL Remote]」681 ページ
- 「sr_time_format オプション [SQL Remote]」682 ページ
- 「sr_timestamp_format [SQL Remote]」683 ページ
- 「SQL Remote オプション」『SQL Remote』


## st_geometry_asbinary_format オプション

ジオメトリからバイナリへの空間データ値の変換方法を制御します。

### 指定可能な値

このオプションでサポートされている値のリストは、ST_AsBinary メソッドで説明されているパラメーター (サブパラメーターも含む) のリストと同じです。ST_Geometry タイプの ST_AsBinary メソッド『SQL Anywhere サーバー 空間データサポート』を参照してください。

### デフォルト

WKB

### 備考

このオプションは、ジオメトリからバイナリへの変換方法を決定するために使用します。次に、このオプションが使用される主なコンテキストを 2 つ示します。

- クライアントアプリケーションから値をバイナリとしてフェッチ。

- CAST 文の実行。例：CAST( geometry-expression AS LONG BINARY ) CAST( geometry-expression AS LONG BINARY )

クライアントアプリケーションからのフェッチと CAST 文以外のコンテキストでは、ジオメトリタイプから BINARY データ型への割り当てによりエラーになります。たとえば、BINARY パラメーターを受け入れるファンクションまたはプロシージャーを呼び出してジオメトリを渡すことはできず、ジオメトリを LONG BINARY 変数に割り当てることもできません。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照

- ST_Geometry タイプの ST_AsBinary メソッド『SQL Anywhere サーバー 空間データサポート』
- 「st_geometry_astext_format オプション」687 ページ
- 「st_geometry_asxml_format オプション」688 ページ
- 「st_geometry_describe_type オプション」689 ページ
- 「st_geometry_on_invalid オプション」691 ページ
- st_geometry_asbinary_format 接続プロパティ745 ページ
- st_geometry_astext_format 接続プロパティ745 ページ
- st_geometry_asxml_format 接続プロパティ745 ページ
- st_geometry_describe_type 接続プロパティ745 ページ
- st_geometry_on_invalid 接続プロパティ745 ページ

## st_geometry_astext_format オプション

ジオメトリからテキストへの空間データ値の変換方法を制御します。

### 指定可能な値

このオプションでサポートされている値のリストは、ST_AsText メソッドで説明されているパラメーター (サブパラメーターも含む) のリストと同じです。ST_Geometry タイプの ST_AsText メソッド『SQL Anywhere サーバー 空間データサポート』を参照してください。

### デフォルト

WKT

### 備考

このオプションは、ジオメトリからテキストへの変換方法を決定するために使用します。次に、このオプションが使用される主なコンテキストを 2 つ示します。

- クライアントアプリケーションから値をテキストとしてフェッチ。
- CAST 文の実行。例：CAST( geometry-expression AS LONG VARCHAR ) CAST( geometry-expression AS LONG VARCHAR )

クライアントアプリケーションからのフェッチと CAST 文以外のコンテキストでは、ジオメトリタイプから CHAR データ型への割り当てによりエラーになります。たとえば、CHAR パラメーターを受け入れるファンクションまたはプロシージャーを呼び出してジオメトリを渡すことはできず、ジオメトリを LONG VARCHAR 変数に割り当てることもできません。

### 参照


- ST_Geometry タイプの ST_AsText メソッド『SQL Anywhere サーバー 空間データサポート』
- 「st_geometry_asbinary_format オプション」686 ページ
- 「st_geometry_asxml_format オプション」688 ページ
- 「st_geometry_describe_type オプション」689 ページ
- 「st_geometry_on_invalid オプション」691 ページ
- st_geometry_asbinary_format 接続プロパティ745 ページ
- st_geometry_astext_format 接続プロパティ745 ページ
- st_geometry_asxml_format 接続プロパティ745 ページ
- st_geometry_describe_type 接続プロパティ745 ページ
- st_geometry_on_invalid 接続プロパティ745 ページ


## st_geometry_asxml_format オプション

ジオメトリから XML への空間データ値の変換方法を制御します。

### 指定可能な値

このオプションでサポートされている値のリストは、ST_AsXML メソッドで説明されているパラメーター (サブパラメーターも含む) のリストと同じです。ST_Geometry タイプの ST_AsXML メソッド『SQL Anywhere サーバー 空間データサポート』を参照してください。

### デフォルト

GML

### 備考

このオプションは、ジオメトリから XML への変換方法を決定するために使用します。次に、このオプションが使用される主なコンテキストを 2 つ示します。

- クライアントアプリケーションから値を XML としてフェッチ。
- CAST 文の実行。例：CAST( geometry-expression AS XML ) CAST( geometry-expression AS XML )

クライアントアプリケーションからのフェッチと CAST 文以外のコンテキストでは、ジオメトリタイプから XML への割り当てによりエラーになります。たとえば、XML パラメーターを受け入れるファンクションまたはプロシージャーを呼び出してジオメトリを渡すことはできず、ジオメトリを XML 変数に割り当てることもできません。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照

- ST_Geometry タイプの ST_AsXML メソッド『SQL Anywhere サーバー 空間データサポート』
- 「st_geometry_asbinary_format オプション」686 ページ
- 「st_geometry_astext_format オプション」687 ページ
- 「st_geometry_describe_type オプション」689 ページ
- 「st_geometry_on_invalid オプション」691 ページ
- st_geometry_asbinary_format 接続プロパティ745 ページ
- st_geometry_astext_format 接続プロパティ745 ページ
- st_geometry_asxml_format 接続プロパティ745 ページ
- st_geometry_describe_type 接続プロパティ745 ページ
- st_geometry_on_invalid 接続プロパティ745 ページ

## st_geometry_describe_type オプション

空間データの記述方法を制御します。

### 指定可能な値

- CHAR
- NCHAR
- BINARY

### デフォルト

CHAR

### 備考

クライアントライブラリでは空間データ型が直接サポートされないため、空間値は文字またはバイナリとしてフェッチされます。st_geometry_describe_type は、ジオメトリカラムまたはジオメトリ式をクライアントに対して記述する方法を指定します。多くのクライアントアプリケーションでは、カラムまたは式の記述タイプを使用して、カラムまたは式のフェッチに使用するクライアントデータ型が決まります。一部のクライアントアプリケーションでは記述タイプが無視されるため、st_geometry_describe_type を設定しても期待する結果にならないことがあります。

ジオメトリを文字列型としてフェッチする場合は、st_geometry_astext_format オプションを使用してジオメトリからテキストへの変換方法が決まります。バイナリ型としてフェッチする場合は、st_geometry_asbinary_format オプションが使用されます。CAST 文の場合にも同じルールが使用されます。

ODBC アプリケーションでは、どのジオメトリタイプのカラムについても、SQLColAttribute の SQL_DESC_TYPE_NAME と SQL_DESC_LOCAL_TYPE_NAME フィールド識別子から文字列 st_geometry が返されます。JDBC では、ResultSetMetaData.getColumnTypeName メソッドによってこれが表現されます。

### 参照


- 「st_geometry_asbinary_format オプション」686 ページ
- 「st_geometry_astext_format オプション」687 ページ
- 「st_geometry_asxml_format オプション」688 ページ
- 「st_geometry_on_invalid オプション」691 ページ
- st_geometry_asbinary_format 接続プロパティ745 ページ
- st_geometry_astext_format 接続プロパティ745 ページ
- st_geometry_asxml_format 接続プロパティ745 ページ
- st_geometry_describe_type 接続プロパティ745 ページ
- st_geometry_on_invalid 接続プロパティ745 ページ


## st_geometry_interpolation オプション

空間計算が円弧を含むときに内挿法を制御します。

### 指定可能な値

このデータベースオプションは、セミコロンで区切られた key=value ペアのセット (key1=value1;key2=value2;...) になります。許可されているキーは、relative-tolerance-percent と absolute-tolerance です。

- **relative-tolerance-percent=*DOUBLE*** relative-tolerance-percent は、最大の変動をパーセントで表記して指定するために使用します。これは、ST_CircularString セグメントの半径に対して、ST_CircularString セグメントの中心点から内挿された ST_LineString セグメントまでの距離で許可されています。relative-tolerance-percent に許可される最小値は 0 であり、absolute-tolerance 設定が常に使用されることを意味します。

- **relative-tolerance-percent=*DOUBLE*** absolute-tolerance は、最大の絶対変動を線形測定単位で表記して指定するために使用します。これは、ST_CircularString セグメントの中心点から内挿された ST_LineString セグメントまでの距離で許可されています。absolute-tolerance に許可される最小値は、SRID で指定される許容値でもあります。

### デフォルト

どちらのキーも指定されない場合、relative-tolerance-percent のデフォルト値は 0.3 になり、これは、完全な円に 40 の点を提供します。また、absolute-tolerance のデフォルト値は SRID で指定される許容値になります。

1 つのキーだけが指定される場合、もう 1 方のキーはその最小値に設定されます。

両方のキーが指定される場合、点の最小数はすべての制限を満たすかたちで生成されます。

### 備考

データベースオプションの設定に関係なく、曲線セグメントごとに少なくとも 3 つの点があります。さらに、内挿された ST_LineString の連続する 2 つの点が互いの許容値内に存在することはありません。

ST_LineString の ST_Envelope がオリジナルの ST_CircularString の ST_Envelope と同じになるように、さらに点を追加することもできます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照

- 「補間が空間の計算に与える影響」『SQL Anywhere サーバー 空間データサポート』
- 「ST_CircularString タイプ」『SQL Anywhere サーバー 空間データサポート』
- 「ST_LineString タイプ」『SQL Anywhere サーバー 空間データサポート』
- ST_Geometry タイプの ST_Envelope メソッド『SQL Anywhere サーバー 空間データサポート』
- 「「グリッドにスナップ」と許容度が空間の計算に与える影響」『SQL Anywhere サーバー 空間データサポート』

## st_geometry_on_invalid オプション

ジオメトリが基本的な検証に失敗した場合 (1 ポイントの LineString、リングが閉じていない多角形、空間参照系の境界を越えたジオメトリなど) の動作を制御します。

### 指定可能な値

- **Ignore**　ジオメトリが表面検証に失敗しても何も処理を行わず、操作を続行します。
- **Error**　ジオメトリが表面検証に失敗するとエラーを返し、操作が失敗します。

### デフォルト

Error

### 備考

無効な空間データをロードする必要がある場合は、オプション値を Ignore に設定してから無効なデータをロードします。データをロードしたら、ST_IsSimple や ST_IsValue などのメソッドを使用して無効なローを判断したり、ST_AsText を使用してデータの WKT 表現を取得したりします。他のほとんどの空間メソッドでは、無効なジオメトリを渡すとエラーになります。テキストエディターを使用して、無効なジオメトリを WKT で修正し、データベースでジオメトリを更新します。

### 参照

- 「st_geometry_asbinary_format オプション」686 ページ
- 「st_geometry_astext_format オプション」687 ページ
- 「st_geometry_asxml_format オプション」688 ページ
- 「st_geometry_describe_type オプション」689 ページ
- st_geometry_asbinary_format 接続プロパティ745 ページ
- st_geometry_astext_format 接続プロパティ745 ページ
- st_geometry_asxml_format 接続プロパティ745 ページ
- st_geometry_describe_type 接続プロパティ745 ページ
- st_geometry_on_invalid 接続プロパティ745 ページ

## string_rtruncation オプション

文字列がトランケートされた場合にエラーを発生させるかどうかを決定します。

### 指定可能な値

On、Off

### デフォルト

On

### 備考

トランケート文字がスペースだけで構成されている場合、例外は発生しません。On に設定すると、ANSI/ISO SQL/2008 の動作に対応します。Off に設定すると、例外は発生せず、文字列は何も通知されずにトランケートされます。

文字列のトランケートは、さまざまな場所で発生する可能性があります。たとえば、INSERT、UPDATE、CAST 文を使用したり、変数に割り当てたりすると、宣言したデータ型が短すぎた場合、文字列がトランケートされる場合があります。

### 参照


- 「文字データ型」『SQL Anywhere サーバー SQL リファレンス』
- 「互換性オプション」584 ページ
- 「SET 文 [T-SQL]」『SQL Anywhere サーバー SQL リファレンス』
- string_rtruncation 接続プロパティ 746 ページ


## subscribe_by_remote オプション [SQL Remote]

SUBSCRIBE BY 値が NULL または空の文字列である場合の解釈を制御します。

### 指定可能な値

On、Off

### デフォルト

On

### 備考

このオプションを On に設定すると、NULL または空の文字列である SUBSCRIBE BY 値のあるローに対してリモートデータベースから操作が行われた場合、リモートユーザーがローにサブスクリプションを作成したと見なします。Off に設定すると、リモートユーザーはローにサブスクリプションを作成していないと見なされます。

このオプションでの唯一の制約は、リモートユーザーが NULL 値のローまたは空のサブスクリプション式 (情報が統合データベースにだけ格納されているもの) を、実際に INSERT (または UPDATE) しようとした場合にエラーが発生することです。この問題はそれほど深刻ではなく、インストール環境で、リモートユーザーに属さないサブスクリプション値を割り当てることで対処できます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照

- 「SQL Remote オプション」『SQL Remote』
- 「多対多の関係における subscribe_by_remote オプション」『SQL Remote』

## subsume_row_locks オプション

データベースサーバーがテーブル用に個別のローロックをいつ取得するかを制御します。

### 指定可能な値

On、Off

### デフォルト

On

### 備考

subsume_row_locks オプションが On (デフォルト) の場合、LOCK TABLE *t* IN EXCLUSIVE MODE で排他的にテーブル *t* がロックされるたびに、データベースサーバーは *t* 用に個別のローロックを取得しなくなります。

これにより、単一のトランザクションで広範な更新が *t* に対して実行された場合 (特に *t* がキャッシュサイズに比べて大きい場合)、パフォーマンスが大幅に改善されます。また、ロックテーブルで現在処理可能な量より多くのアトミック更新オペレーションを実行できるようになります (約 200 万～ 400 万ロー)。

このオプションが On のときに、テーブル上のキーセットカーソルがこのような形でロックされると、データベース内のいずれかのローが変更される場合にカーソル内のすべてのローに対してロー変更警告が返されます。データベースサーバーは ORDER BY のある更新可能なカーソルを、結果としてキーセットカーソルにすることができます。

### 参照

- 「LOCK TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- subsume_row_locks 接続プロパティ 746 ページ

## suppress_tds_debugging オプション

TDS デバッグ情報をデータベースサーバーメッセージウィンドウに表示するかどうかを決定します。

### 指定可能な値

On、Off

### デフォルト

Off

### 備考

データベースサーバーを -z オプションで起動すると、TDS プロトコルに関するデバッグ情報を含むデバッグ情報がデータベースサーバーメッセージウィンドウに表示されます。

suppress_tds_debugging オプションは、TDS に関するデバッグ情報がデータベースサーバーメッセージウィンドウに表示されないようにします。このオプションを Off (デフォルト) に設定すると、TDS デバッグ情報がデータベースサーバーメッセージウィンドウに表示されます。

### 参照


- 「-z dbeng12/dbsrv12 サーバーオプション」296 ページ
- suppress_tds_debugging 接続プロパティ746 ページ


## synchronize_mirror_on_commit オプション

非同期モードまたは非同期フルページモードにおいて、データベースの変更がミラーサーバーへ送信されたことがどの時点で保証されるかを制御します。

### 指定可能な値

On、Off

### デフォルト

Off

### 備考

synchronize_mirror_on_commit オプションでは、非同期モードまたは非同期フルページモードにおいて、データベースの変更がミラーサーバーへ送信されたことがどの時点で保証されるかについて厳密に制御できます。このオプションのデフォルトは Off です。On に設定すると、COMMIT が発生するたびにトランザクションログに記録されたすべての変更がミラーサーバーへ送信され、それらの変更がミラーサーバーで受信されると、ミラーサーバーからプライマリサーバーへ受信確認が送信されます。このオプションは、SET TEMPORARY OPTION を使用して特定のトランザクションに対して設定できます。また、ログインプロシージャーで APPINFO 文字列を調べ、このオプションを特定のアプリケーションだけに設定すると、効果的な場合もあります。これによって、ミラーリング動作をさまざまなアプリケーションのニーズに適合させることができます。

### 参照


- 「データベースミラーリング」1137 ページ
- synchronize_mirror_on_commit 接続プロパティ746 ページ


## tds_empty_string_is_null オプション

TDS 接続で空の文字を NULL として返すか、ブランク文字 1 文字を含む文字列として返すかを制御します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 指定可能な値

On、Off

### デフォルト

Off

### 備考

デフォルトでは、このオプションには Off が設定され、空の文字列は、TDS 接続用の 1 文字のブランク文字を含む文字列で返されます。このオプションを On に設定すると、空の文字列は、TDS 接続では NULL 文字列として返されます。TDS 以外の接続では、空の文字列と NULL 文字列は区別されます。

### 参照

- 「NULL 特別値」『SQL Anywhere サーバー SQL リファレンス』
- tds_empty_string_is_null 接続プロパティ746 ページ

## temp_space_limit_check オプション

接続によって使用されるテンポラリファイル領域のサイズを確認し、要求された領域サイズが接続に許容されるサイズよりも大きい場合は要求は失敗になります。

### 指定可能な値

On、Off

### デフォルト

On

### スコープ

PUBLIC グループのみに設定できます。DBA 権限が必要です。

### 備考

temp_space_limit_check オプションを On に設定すると、接続がそのクォータを超えるテンポラリファイル領域を要求した場合、その要求は失敗し、エラー SQLSTATE_TEMP_SPACE_LIMIT が返されます。このオプションを Off (デフォルト) に設定すると、データベースサーバーは接続が使用するテンポラリファイル領域のサイズをチェックしません。接続がそのクォータを超えるテンポラリ領域を要求した場合、このオプションが Off に設定されていると、致命的なエラーが発生することがあります。

次の 2 つのしきい値のうち小さい方が、接続のテンポラリファイルのクォータになります。

1. max_temp_space オプションで指定された各接続に許可されるテンポラリファイル領域の最大サイズ

2. テンポラリファイルのサイズを接続数で除算した見込みの最大サイズ

しきい値は、テンポラリファイルのサイズが最大サイズの 80% に達した場合にのみ使用されます。これは、オペレーティングシステムによって報告される、デバイス上に残っている空き領域のサイズに基づいて判断されます。接続要求によって、クォータが許可するサイズよりも大きいテンポラリファイルが要求された場合、接続の現在の要求は SQLSTATE 54W05 (TEMP_SPACE_LIMIT) で失敗します。

接続が使用するテンポラリファイル領域の最大サイズは、max_temp_space オプションで指定できます。

### 参照


- 「sa_disk_free_space システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「max_temp_space オプション」647 ページ
- 「-dt dbeng12/dbsrv12 サーバーオプション」221 ページ
- temp_space_limit_check 接続プロパティ 747 ページ
- 「SQL Anywhere のサイズと数の制限」811 ページ


## time_format オプション

データベースから取り出した時刻のフォーマットを設定します。

### 指定可能な値

文字列 (次の記号の組み合わせ)

### デフォルト

HH:NN:SS.SSS

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

### 備考

フォーマットは次の記号を組み合わせた文字列です。

| 記号 | 説明 |
|---|---|
| HH | 2 桁の時間 |
| NN | 2 桁の分 |
| SS.SSSSSS | 秒と小数点第 6 位までの秒の小数。小数点以下 6 桁のタイムスタンプをサポートしないプラットフォームもあります。 |
| AA | A.M. または P.M. (12 時間表記) − 24 時間表記の場合、AA と PP は省略します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 記号 | 説明 |
| --- | --- |
| PP | P.M. (必要に応じて) (12 時間表記) – 24 時間表記の場合、AA と PP は省略します。 |

各記号は、フォーマットしようとする日付のデータで置き換えられます。

数値データを表す記号では、記号に大文字を使用するか小文字を使用するかで 0 埋め込みを制御できます。

- 記号をすべて大文字または小文字 (NN や nn など) で入力すると、0 埋め込みが行われます。たとえば hh:nn:ss.sss と入力すると、02:34:56.123 と表記されます。

- 大文字と小文字を混ぜて入力すると (Nn など)、0 の埋め込みは行われません。たとえば Hh:nn:ss.sss と入力すると、2:34:56.123 と表記されます。

- 小数点以下の秒の最初の 2 桁の大文字と小文字を混ぜて入力すると (Ss または sSssss)、後ろの 0 が削除されます。たとえば hh:nn:ss.Sss と入力すると、12:34:56.1 と表記されます。

### 参照


- 「date_format オプション」613 ページ
- 「timestamp_format オプション」698 ページ
- 「timestamp_with_time_zone_format オプション」700 ページ
- 「互換性オプション」584 ページ
- 「sp_tsql_environment システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「日付と時刻のデータベースからの取得」『SQL Anywhere サーバー SQL リファレンス』
- time_format 接続プロパティ 747 ページ


## time_zone_adjustment オプション

1 つの接続でタイムゾーンの調整を修正できます。

### 指定可能な値

符号なし整数 (例： 300)

引用符で囲まれた正または負の符号付き整数 (例：'+300' または '-300')

先頭が + または - で、引用符で囲まれた時と分を表す文字列 (例： '+5:00'、'-5:00')

### デフォルト

クライアントが Embedded SQL、ODBC、OLE DB、ADO、または ADO.NET を介して接続している場合は、クライアントのタイムゾーンに従ってデフォルト値が設定されます。クライアントが jConnect または Sybase Open Client を介して接続している場合は、デフォルトは、データベースサーバーのタイムゾーンに基づいて設定されます。

### 備考

time_zone_adjustment オプションの値は SELECT CONNECTION_PROPERTY( 'TimeZoneAdjustment' ) が返す値と同じです。この値は、接続のローカル時間を表示するために協定世界時 (UTC: Coordinated Universal Time) に加算する必要がある分数を表します。

このオプションで TIMESTAMP から TIMESTAMP WITH TIME ZONE への変換を制御する方法については、TIMESTAMP WITH TIME ZONE との変換『SQL Anywhere サーバー SQL リファレンス』を参照してください。

### 参照

- time_zone_adjustment 接続プロパティ 747 ページ

## timestamp_format オプション

データベースから取り出したタイムスタンプのフォーマットを設定します。

### 指定可能な値

文字列 (次の記号の組み合わせ)

### デフォルト

YYYY-MM-DD HH:NN:SS.SSS

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

### 備考

フォーマットは次の記号を組み合わせた文字列です。

| 記号 | 説明 |
|---|---|
| YY | 2 桁の年 |
| YYYY | 4 桁の年 |
| MM | 2 桁の月、または 2 桁の分 (HH:MM のように、コロンの後ろに続く場合) |
| MMM[m...] | 月を略した文字、"m" の数だけ文字を使用 |
| DD | 2 桁の日 |
| DDD[d...] | 曜日を略した文字 |
| HH | 2 桁の時間 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 記号 | 説明 |
| --- | --- |
| NN | 2 桁の分 |
| SS.SSSSSS | 秒と小数点第 6 位までの秒の小数。小数点以下 6 桁のタイムスタンプをサポートしないプラットフォームもあります。 |
| AA | A.M. または P.M. (12 時間表記) – 24 時間表記の場合、AA と PP は省略します。 |
| PP | P.M. (必要に応じて) (12 時間表記) – 24 時間表記の場合、AA と PP は省略します。 |

各記号は、フォーマットされる日付のデータで置き換えられます。

文字データを表す記号 (*MMM* など) では、出力の文字を次のように制御できます。

- 記号をすべて大文字で入力すると、フォーマットはすべて大文字で表記されます。たとえば、MMM と入力すると、JAN と表記されます。
- 記号をすべて小文字で入力すると、フォーマットはすべて小文字で表記されます。たとえば、mmm と入力すると、jan と表記されます。
- 大文字と小文字を混ぜて入力すると、使用される言語に適切な文字が SQL Anywhere により選択されます。たとえば、Mmm と入力すると、英語では May、フランス語では mai と表記されます。

文字データがマルチバイトである場合、各記号の長さはバイト数ではなく文字数を表します。たとえば 'mmm' は 3 文字の月名を示しています。

数値データを表す記号では、記号に大文字を使用するか小文字を使用するかで 0 埋め込みを制御できます。

- 記号をすべて大文字または小文字 (MM や mm など) で入力すると、0 埋め込みが行われます。たとえば、yyyy/mm/dd と入力すると、2002/01/01 と表記されます。
- 大文字と小文字を混ぜて入力すると (Mm など)、0 の埋め込みは行われません。たとえば、yyyy/Mm/Dd と入力すると、2002/1/1 と表記されます。
- 小数点以下の秒の最初の 2 桁の大文字と小文字を混ぜて入力すると (Ss または sSssss)、後ろの 0 が削除されます。たとえば hh:nn:ss.Sss と入力すると、12:34:56.1 と表記されます。

**注意**
日付フォーマットの順序を変更するように timestamp_format の設定を変更する場合は、同じ変更を反映するように date_order オプションも変更し、同様に date_order を変更する場合は timestamp_format も変更してください。「date_order オプション」615 ページを参照してください。

### 参照


- 「date_format オプション」613 ページ
- 「time_format オプション」696 ページ
- 「timestamp_with_time_zone_format オプション」700 ページ
- 「互換性オプション」584 ページ
- 「sp_tsql_environment システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- timestamp_format 接続プロパティ 747 ページ


## timestamp_with_time_zone_format オプション

データベースから取り出した TIMESTAMP WITH TIME ZONE 値のフォーマットを設定します。

### 指定可能な値

文字列 (次の記号の組み合わせ)

### デフォルト

YYYY-MM-DD HH:NN:SS.SSS+HH:NN

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

### 備考

フォーマットは次の記号を組み合わせた文字列です。

| 記号 | 説明 |
|---|---|
| YY | 2 桁の年 |
| YYYY | 4 桁の年 |
| MM | 2 桁の月、または 2 桁の分 (HH:MM のように、コロンの後ろに続く場合) |
| MMM[m...] | 月を略した文字、"m" の数だけ文字を使用 |
| DD | 2 桁の日 |
| DDD[d...] | 曜日を略した文字 |
| HH | 2 桁の時間 |
| NN | 2 桁の分 |
| SS.SSSSSS | 秒と小数点第 6 位までの秒の小数。小数点以下 6 桁のタイムスタンプをサポートしないプラットフォームもあります。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 記号 | 説明 |
|---|---|
| AA | A.M. または P.M. (12 時間表記) – 24 時間表記の場合、AA と PP は省略します。 |
| PP | P.M. (必要に応じて) (12 時間表記) - 24 時間表記の場合、AA と PP は省略します。 |
| HH | 2 桁の時間 (タイムゾーンオフセット) |
| NN | 2 桁の分 (タイムゾーンオフセット) |

各記号は、フォーマットされる日付のデータで置き換えられます。

文字データを表す記号 (*MMM* など) では、出力の文字を次のように制御できます。

- 記号をすべて大文字で入力すると、フォーマットはすべて大文字で表記されます。たとえば、MMM と入力すると、JAN と表記されます。

- 記号をすべて小文字で入力すると、フォーマットはすべて小文字で表記されます。たとえば、mmm と入力すると、jan と表記されます。

- 大文字と小文字を混ぜて入力すると、使用される言語に適切な文字が SQL Anywhere により選択されます。たとえば、Mmm と入力すると、英語では May、フランス語では mai と表記されます。

文字データがマルチバイトである場合、各記号の長さはバイト数ではなく文字数を表します。たとえば、mmm は 3 文字の月名を示しています。

数値データを表す記号では、記号に大文字を使用するか小文字を使用するかで 0 埋め込みを制御できます。

- 記号をすべて大文字または小文字 (MM や mm など) で入力すると、0 埋め込みが行われます。たとえば、yyyy/mm/dd と入力すると、2002/01/01 と表記されます。

- 大文字と小文字を混ぜて入力すると (Mm など)、0 の埋め込みは行われません。たとえば、yyyy/Mm/Dd と入力すると、2002/1/1 と表記されます。

- 小数点以下の秒の最初の 2 桁の大文字と小文字を混ぜて入力すると (Ss または sSssss)、後ろの 0 が削除されます。たとえば hh:nn:ss.Sss と入力すると、12:34:56.1 と表記されます。

**注意**
日付フォーマットの順序を変更するように timestamp_with_time_zone_format オプションの設定を変更する場合は、同じ変更を反映するように date_order オプションも変更し、同様に date_order を変更する場合は timestamp_format も変更してください。「date_order オプション」615 ページを参照してください。

### 参照


- 「TIMESTAMP WITH TIME ZONE データ型」『SQL Anywhere サーバー SQL リファレンス』
- 「date_format オプション」613 ページ
- 「time_format オプション」696 ページ
- 「timestamp_format オプション」698 ページ


## truncate_timestamp_values オプション

TIMESTAMP 値の精度を制限します。

### 指定可能な値

On、Off

### デフォルト

Off

### スコープ

PUBLIC グループのみに設定できます。DBA 権限が必要です。このオプションは、タイムスタンプデータがすでに格納されているデータベースに対しては有効にしないでください。

### 備考

SQL Anywhere の TIMESTAMP 値の精度は、小数点以下 6 桁です。ただし、TIMESTAMP 値を小数点以下 3 桁などにトランケートする他のソフトウェアとの互換性を維持するために、truncate_timestamp_values オプションを On に設定すると、SQL Anywhere が格納する小数点以下の桁数を制限できます。default_timestamp_increment オプションは、TIMESTAMP 値を小数点以下何桁でトランケートするかを決定します。

Mobile Link 同期に対し、このオプションを設定する場合は、最初の同期の実行前に設定する必要があります。

データベースサーバーが truncate_timestamp_values と default_timestamp_increment の組み合わせで指定されたものより細かい TIMESTAMP 値を検出した場合、エラーがレポートされます。

通常、データベースをアンロードしてそれを truncate_timestamp_values 値と default_timestamp_increment 値が設定されている新しいデータベースに再ロードする方法が、確実に適切な TIMESTAMP 値を使用する一番簡単なソリューションです。ただし、テーブル内の TIMESTAMP カラムのタイプにより、次のように行うこともできます。

- TIMESTAMP カラムが DEFAULT TIMESTAMP または DEFAULT UTC TIMESTAMP で定義されている場合 (つまり、ローが変更されるとデータベースサーバーによって値が自動的に更新される)、truncate_timestamp_values オプションを変更する前にテーブル内のすべてのローを削除する必要があります。ローは DELETE 文または TRUNCATE TABLE 文を使用して削除します。

- TIMESTAMP カラムが DEFAULT TIMESTAMP または DEFAULT UTC TIMESTAMP 以外で定義されている場合、UPDATE 文を実行して文字列に値を割り当てて、TIMESTAMP に戻します。次に例を示します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
UPDATE T
  SET ts = CAST( DATEFORMAT( ts, 'yyyy/mm/dd hh:nn:ss.ss' )
 AS TIMESTAMP );
```

この手順は必要とされる精度より低下する可能性があります。使用するフォーマット文字列は、保持する精度の桁数によります。

### 参照

- 「default_timestamp_increment オプション」618 ページ
- 「TIMESTAMP 特別値」『SQL Anywhere サーバー SQL リファレンス』
- 「CURRENT UTC TIMESTAMP 特別値」『SQL Anywhere サーバー SQL リファレンス』
- 「UTC TIMESTAMP 特別値」『SQL Anywhere サーバー SQL リファレンス』
- truncate_timestamp_values 接続プロパティ748 ページ

### 例

たとえば、default_timestamp_increment オプションを 100000 に設定すると、秒の部分は小数点以下 1 桁の後でトランケートされるので、"2000/12/05 10:50:53.700" のような値を格納できるようになります。

## tsql_outer_joins オプション

Transact-SQL 外部ジョイン演算子の *= と =* を文やビューで使用可能にするかどうかを指定します。

### 指定可能な値

On、Off

### デフォルト

Off

### 備考

Transact-SQL 外部ジョインはサポートされなくなりました。このオプションを On に設定すると、Transact-SQL 外部ジョインを使用できます。

### 参照

- 「互換性オプション」584 ページ
- 「Transact-SQL の外部ジョイン (*= or =*)」『SQL Anywhere サーバー SQL の使用法』
- 「ジョイン演算子」『SQL Anywhere サーバー SQL リファレンス』
- sql_outer_joins 接続プロパティ748 ページ

## tsql_variables オプション

@ 符号を、Embedded SQL ホスト変数名のプレフィクスとして使用できるかどうかを制御します。

### 指定可能な値

On、Off

### デフォルト

Off

Sybase Open Client 接続と jConnect 接続の場合は On

### 備考

このオプションを On に設定すると、Embedded SQL のホスト変数名のプレフィクスとしてコロンの代わりに @ 符号を使用できます。これは、主に Transact-SQL との互換性を保つために実装されています。

### 参照


- 「互換性オプション」584 ページ
- 「sp_tsql_environment システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- tsql_variables 接続プロパティ748 ページ


## updatable_statement_isolation オプション

isolation_level オプションが Readonly-statement-snapshot に設定されている場合に更新可能な文に適用する独立性レベルを指定します。

### 指定可能な値

0, 1, 2, 3

### デフォルト

0

### 備考

updatable_statement_isolation オプションで指定した独立性レベルは、isolation_level オプションが Readonly-statement-snapshot に設定されている場合に、更新可能な文によって使用されます。次の値を指定できます。

- **0** ダーティリード、繰り返し不可能読み出し、幻ローを許可します。
- **1** ダーティリードを防ぎます。繰り返し不可能読み出しと幻ローを許可します。
- **2** ダーティリードと繰り返し不可能読み出しを防ぎます。幻ローを許可します。
- **3** 直列化可能。ダーティリード、繰り返し不可能読み出し、幻ローを防ぎます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照

- 「isolation_level オプション」630 ページ
- 「スナップショットアイソレーション」『SQL Anywhere サーバー SQL の使用法』
- 「独立性レベルと一貫性」『SQL Anywhere サーバー SQL の使用法』
- 「独立性レベル選択のガイドライン」『SQL Anywhere サーバー SQL の使用法』
- 「典型的な矛盾のケース」『SQL Anywhere サーバー SQL の使用法』
- updatable_statement_isolation 接続プロパティ748 ページ

## update_statistics オプション

接続から統計ガバナーにクエリのフィードバックを送信できるかどうかを制御します。

### 指定可能な値

On、Off

### デフォルト

On

### 備考

データベースサーバーは通常のクエリ実行中に統計情報を収集し、収集した統計を使用してカラム統計を自己チューニングします。このオプションを Off に設定すると、統計ガバナーは統計の正常性に関するフィードバックを受信せず、指定された接続の統計も修正しません。ただし、統計の使用状況のモニターと、使用状況に基づいた統計の作成または削除は引き続き行われます。

DBA 権限を持つユーザーがこのオプションを設定した場合は、設定内容がすべてのデータベース接続に影響します。そうでない場合は、現在のユーザーの接続のみが影響を受けます。

通常の環境において、このオプションをオフにする必要はありません。

update_statistics オプションは、データの更新 (LOAD/INSERT/UPDATE/DELETE) による統計の変更に影響しません。これらの文に基づいて統計を更新するかどうかを設定するには、collect_statistics_on_dml_updates データベースオプションを使用します。

collect_statistics_on_dml_updates オプションと update_statistics オプションが異なる点は、update_statistics オプションを On に設定した場合、ある述部を満たすローの実際の数とその述部を満たすと予想されるローの数が比較されてから、推定値が更新されることです。collect_statistics_on_dml_updates オプションを On に設定した場合は、挿入、更新、または削除されたローの値に基づいてカラム統計が修正されます。

### 参照

- 「collect_statistics_on_dml_updates オプション」606 ページ
- 「統計ガバナーが統計を保守する方法」『SQL Anywhere サーバー SQL の使用法』
- 「カラム統計の更新」『SQL Anywhere サーバー SQL の使用法』
- update_statistics 接続プロパティ748 ページ

# user_estimates オプション

クエリの述部に含まれるユーザー選択性推定をクエリオプティマイザーが尊重するか無視するかを制御します。

### 指定可能な値

Enabled、Disabled、Override-Magic

### デフォルト

Override-Magic

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

### 備考

SQL Anywhere では、ユーザー選択性推定を指定することで、データベースサーバーが述部の選択性を正確に推定できないときにオプティマイザーのパフォーマンスを向上させることができます。ただし、ユーザー選択性推定は、適切な状況でのみ使用してください。たとえば、オプティマイザーが使用する Override-Magic 選択性推定と実際の選択性が大きく異なっている場合は、1 つ以上の関数が関係する述部に対して選択性推定を指定することは有用です。

ソフトウェアによって選択されたアクセスプランが不適切であり、パフォーマンス問題を回避するために選択性推定を使用したものの、それが不正確であった場合は、このオプションを Disabled に設定することをおすすめします。不正確な推定が使用された場合、データベースサーバーは最適なプランを選択できません。

ユーザー選択性推定の詳細については、「明示的な選択性推定」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

ユーザー選択性推定に述部が指定されているときは、このオプションの設定に基づいて、その推定は尊重されるか無視されます。次の値を指定できます。

- **Enabled**　ユーザーが提供する選択性推定をすべて尊重します。このオプションは、On を使用して有効にすることもできます。
- **Override-Magic**　ユーザー選択性推定は尊重されますが、オプティマイザーが最後の手段であるヒューリスティック値 (マジック値とも呼ばれます) の使用を選択する以外に方法がない場合のみ使用されます。
- **Disabled**　他の推定データが使用できないときは、ユーザー推定は無視され、マジック値が使用されます。このオプションは、Off を使用して無効にすることもできます。

INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT、INTERSECT の各文に OPTION 句を含めることによって、各文で指定したオプション設定をこれらのオプションに対するテンポラリ設定やパブリック設定よりも優先させることができます。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照

- 「INSERT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「UPDATE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「DELETE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「SELECT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「UNION 文」『SQL Anywhere サーバー SQL リファレンス』
- 「EXCEPT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「INTERSECT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「明示的な選択性推定の提供を控える」『SQL Anywhere サーバー SQL の使用法』
- 「明示的な選択性推定」『SQL Anywhere サーバー SQL リファレンス』
- user_estimates 接続プロパティ749 ページ

## uuid_has_hyphens オプション

ユニークな識別子の値が文字列に変換するときのフォーマットを設定します。

### 指定可能な値

On、Off

### デフォルト

On

### 備考

uid_has_hyphens オプションを On に設定すると、変換後の文字列には 4 つのハイフンが挿入されます。たとえば、12345678-1234-1234-1234-1234567890ab のようになります。このオプションを Off に設定すると、文字列にハイフンは挿入されません。文字列からユニークな識別子の値への変換では、データベースサーバーはハイフンが挿入された UUID 文字列とハイフンのない UUID 文字列の両方をサポートします。

### 参照

- 「UNIQUEIDENTIFIER データ型」『SQL Anywhere サーバー SQL リファレンス』

## verify_all_columns オプション [SQL Remote]

ローカルデータベースで発行した更新を含むメッセージがすべてのカラム値を含んで送信されるかどうかを制御します。

### 指定可能な値

On、Off

### デフォルト

Off

### 備考

このオプションは、SQL Remote のみで使用します。On に設定すると、ローカルデータベースの発行した更新を含むメッセージはすべてのカラム値を含んで送信され、カラムの重複はサブスクライバーデータベースで RESOLVE UPDATE トリガーを起動します。

### 参照

- 「SQL Remote オプション」『SQL Remote』
- 「verify_all_columns オプション」『SQL Remote』

### 例

次の文は、SQL Anywhere で、すべてのユーザーに対して verify_all_columns オプションを Off に設定します。

```
SET OPTION PUBLIC.verify_all_columns = 'Off';
```

## verify_password_function オプション

パスワード規則を実装します。

### 指定可能な値

文字列

### デフォルト

空の文字列 (パスワードが設定されていない場合、関数は呼び出されない)

### スコープ

DBA 権限が必要です。

### 備考

verify_password_function で指定された関数は、NULL 以外のパスワードが作成されるか設定されると、自動的に呼び出されます。ユーザーによって関数が上書きされないようにするには、オプション値を *owner.function-name* に設定します。ユーザーは、データベースに接続可能なパスワードを持っている必要があります。

パスワードの作成または設定に使用する文の検証後、関数が呼び出され、指定した規則を使用してパスワードが検証されます。パスワードが規則に従っている場合、関数は成功を示す NULL を返し、呼び出し元の文が実行されます。パスワードが規則に違反している場合は、エラーが設定されるか、エラーを示す NULL 以外の文字列が返されます。NULL 以外の文字列は、パスワードが不合格となった理由として、ユーザーに返されるエラーに追加されます。

パスワード検証関数は、*user_name* VARCHAR(128) と *new_pwd* VARCHAR(255) という 2 つのパラメーターを取ります。戻り値のデータ型は VARCHAR(255) です。パスワード検証関数がデバッガーによってステップスルーされないように、この関数で ALTER FUNCTION *function-name* SET HIDDEN 文を実行することをおすすめします。verify_password_function オプションを設定する場合、GRANT CONNECT 文で複数のユーザー ID とパスワードを指定することはできません。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

●ユーザー ID には、次に該当する値を指定できません。

○ 空白スペース、一重引用符、または二重引用符で始まる値
○ 空白スペースで終わる値
○ セミコロンを含む値

●パスワードは大文字と小文字が区別され、次に該当する値は指定できません。

○ 空白スペース、一重引用符、または二重引用符で始まる値
○ 空白スペースで終わる値
○ セミコロンを含む値
○ 256 バイト長を超える値

パスワード規則の詳細については、「パスワード検証の使用」1384 ページを参照してください。

## 参照


- 「min_password_length オプション」648 ページ
- 「CREATE USER 文」『SQL Anywhere サーバー SQL リファレンス』
- 「CREATE FUNCTION 文」『SQL Anywhere サーバー SQL リファレンス』
- 「ALTER FUNCTION 文」『SQL Anywhere サーバー SQL リファレンス』
- 「ALTER USER 文」『SQL Anywhere サーバー SQL リファレンス』
- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「パスワードのセキュリティ」1383 ページ
- 「NewPassword (NEWPWD) 接続パラメーター」355 ページ
- verify_password_function 接続プロパティ752 ページ


## 例

次の例では、テーブルと関数を 1 つずつ定義し、いくつかのログインポリシーオプションを設定しています。これらは共に、パスワードに特定の種類の文字が含まれることを要求し、パスワードの再利用を禁止して、パスワード有効期限を適用するなど、詳細なパスワード規則を実装します。ユーザー ID が作成されるか、パスワードが変更されると、データベースサーバーによって verify_password_function オプションを使用して関数が呼び出されます。アプリケーションは、post_login_procedure オプションで指定されたプロシージャーを呼び出して、パスワードが期限切れになる前にパスワードの変更が必要であることを通知できます。

このサンプルコードは、*%SQLANYSAMP12%¥SQLAnywhere¥SQL¥verify_password.sql* でも利用できます。

```
-- only DBA should have permissions on this table
CREATE TABLE DBA.t_pwd_history(
      pk          INT         DEFAULT AUTOINCREMENT PRIMARY KEY,
      user_name   CHAR(128),  -- the user whose password is set
      pwd_hash    CHAR(32) ); -- hash of password value to detect
                                -- duplicate passwords

-- called whenever a non-NULL password is set
-- to verify the password conforms to password rules
CREATE FUNCTION DBA.f_verify_pwd( uid     VARCHAR(128),
                                  new_pwd VARCHAR(255) )
RETURNS VARCHAR(255)
BEGIN
   -- a table with one row per character in new_pwd
```

```
   DECLARE local temporary table pwd_chars(
         pos INT PRIMARY KEY,    -- index of c in new_pwd
         c   CHAR( 1 CHAR ) );   -- character
   -- new_pwd with non-alpha characters removed
   DECLARE pwd_alpha_only      CHAR(255);
   DECLARE num_lower_chars     INT;

   -- enforce minimum length (can also be done with
   -- min_password_length option)
   IF length( new_pwd ) < 6 THEN
      RETURN 'password must be at least 6 characters long';
   END IF;

   -- break new_pwd into one row per character
   INSERT INTO pwd_chars SELECT row_num, substr( new_pwd, row_num, 1 )
                         FROM dbo.RowGenerator
                         WHERE row_num <= length( new_pwd );

   -- copy of new_pwd containing alpha-only characters
   SELECT list( c, '' ORDER BY pos ) INTO pwd_alpha_only
      FROM pwd_chars WHERE c BETWEEN 'a' AND 'z' OR c BETWEEN 'A' AND 'Z';

   -- number of lowercase characters IN new_pwd
   SELECT count(*) INTO num_lower_chars
      FROM pwd_chars WHERE CAST( c AS BINARY ) BETWEEN 'a' AND 'z';

   -- enforce rules based on characters contained in new_pwd
   IF ( SELECT count(*) FROM pwd_chars WHERE c BETWEEN '0' AND '9' )
         < 1 THEN
      RETURN 'password must contain at least one numeric digit';
   ELSEIF length( pwd_alpha_only ) < 2 THEN
      RETURN 'password must contain at least two letters';
   ELSEIF num_lower_chars = 0
         OR length( pwd_alpha_only ) - num_lower_chars = 0 THEN
      RETURN 'password must contain both upper- and lowercase characters';
   END IF;

   -- not the same as any user name
   -- (this could be modified to check against a disallowed words table)
   IF EXISTS( SELECT * FROM SYS.SYSUSER
                WHERE lower( user_name ) IN ( lower( pwd_alpha_only ),
                                              lower( new_pwd ) ) ) THEN
      RETURN 'password or only alphabetic characters in password ' ||
            'must not match any user name';
   END IF;

   -- not the same as any previous password for this user
   IF EXISTS( SELECT * FROM t_pwd_history
                WHERE user_name = uid
                  AND pwd_hash = hash( uid || new_pwd, 'md5' ) ) THEN
      RETURN 'previous passwords cannot be reused';
   END IF;

   -- save the new password
   INSERT INTO t_pwd_history( user_name, pwd_hash )
      VALUES( uid, hash( uid || new_pwd, 'md5' ) );

   RETURN( NULL );
END;

ALTER FUNCTION DBA.f_verify_pwd SET HIDDEN;
GRANT EXECUTE ON DBA.f_verify_pwd TO PUBLIC;
SET OPTION PUBLIC.verify_password_function = 'DBA.f_verify_pwd';
```

```
-- All passwords expire in 180 days. Expired passwords can be changed
-- by the user using the NewPassword connection parameter.
ALTER LOGIN POLICY DEFAULT password_life_time = 180;

-- If an application calls the procedure specified by the
-- post_login_procedure option, then the procedure can be used to
-- warn the user that their password is about to expire. In particular,
-- Interactive SQL and Sybase Central call the post_login_procedure.
ALTER LOGIN POLICY DEFAULT password_grace_time = 30;

-- Five consecutive failed login attempts will result in a non-DBA
-- user ID being locked.
ALTER LOGIN POLICY DEFAULT max_failed_login_attempts = 5;
```

## verify_threshold オプション [SQL Remote]

更新がレプリケートされるときにどのカラムが確認されるかを制御します。

### 指定可能な値

整数 (バイト)

### デフォルト

1000

### 備考

このオプションは、SQL Remote のみで使用します。カラムのデータ型がしきい値より長い場合は、カラムの古い値は UPDATE がレプリケートされるときに確認されません。このようにして、SQL Remote メッセージのサイズが大きくならないようにしますが、競合する長い値の更新が検出されないという短所もあります。

### 参照

- 「SQL Remote オプション」『SQL Remote』
- 「BLOB」『SQL Remote』

## wait_for_commit オプション

データが操作されるときに、いつ外部キー整合性をチェックするかを決定します。

### 指定可能な値

On、Off

### デフォルト

Off

### スコープ

個々の接続または PUBLIC グループに設定できます。すぐに有効になります。

### 備考

このオプションを On に設定すると、データベースは次の COMMIT 文まで外部キー整合性をチェックしません。Off の場合は、check_on_commit オプションで作成されていないすべての外部キーが、挿入、更新、削除時にチェックされます。

### 参照

- 「sa_check_commit システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「COMMIT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「挿入時のロック」『SQL Anywhere サーバー SQL の使用法』
- 「参照整合性の検査」『SQL Anywhere サーバー SQL の使用法』
- wait_for_commit 接続プロパティ752 ページ

## webservice_namespace_host オプション

DISH サービスの XML ネームスペース仕様内でホスト名を指定します。

### 指定可能な値

NULL または hostname-string

### デフォルト

NULL

### スコープ

PUBLIC グループのみに設定できます。すぐに有効になります。DBA 権限が必要です。

### 備考

このオプションは、HTTP Host 要求ヘッダーのデフォルトの使用法より優先されます。

Webservices Description Language Documents (WSDL) は、DISH サービスによってエクスポートされます。WSDL は、使用可能な SOAP サービスの説明を含んだ XML ドキュメントです。XML ドキュメント内の targetNameSpace と soapAction 操作の URL にはホスト名が含まれます。このオプションがデフォルト値の NULL に設定されている場合、ホスト名はデータベースサーバー実行中のコンピューターのホスト名になります。このオプションが文字列値に設定されていると、ホスト名にはその文字列が使用されます。このオプションは、配備後は開発に使用したホスト以外のホストを対象とする Web サービスクライアントアプリケーションを開発するときに使用することを目的としています。

### 参照

- 「HTTP Web サーバーとしての SQL Anywhere」『SQL Anywhere サーバー プログラミング』
- webservice_namespace_host 接続プロパティ753 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

# 接続、データベース、データベースサーバーのプロパティ

## 接続プロパティ値のアクセス

接続プロパティは、SQL Anywhere データベースへの各接続で使用できます。CONNECTION_PROPERTY システム関数を使用して個々のプロパティの値を取得するか、sa_conn_properties システムプロシージャーを使用してすべての接続プロパティの値を取得できます。プロパティ名では大文字と小文字が区別されません。

**♦ 接続プロパティの値を取り出すには、次の手順に従います。**

● CONNECTION_PROPERTY システム関数を使用します。

たとえば、次の文は、現在の接続によってファイルから読み込まれたページ数を返します。

```
SELECT CONNECTION_PROPERTY ( 'DiskRead' );
```

**♦ すべての接続プロパティの値を取り出すには、次の手順に従います。**

● sa_conn_properties システムプロシージャーを使用します。

```
CALL sa_conn_properties( );
```

接続プロパティごとに個別のローが表示されます。

**参照**


- 「CONNECTION_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』
- 「sa_conn_activity システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「データベースサーバープロパティ値のアクセス」753 ページ
- 「データベースプロパティ値のアクセス」781 ページ


**接続プロパティ**

| プロパティ名 | 説明 |
|---|---|
| **allow_nulls_by_default** | NULL と NOT NULL のいずれも指定せずに作成されたカラムについて NULL 値の入力を許可するかどうかを示す値を返します。「allow_nulls_by_default オプション」589 ページを参照してください。 |
| **allow_read_client_file** | データベースサーバーがクライアントコンピューター上のファイルの読み込みを許可するかどうかを示す値を返します。「allow_read_client_file オプション」590 ページを参照してください。 |

| プロパティ名 | 説明 |
| --- | --- |
| **allow_snapshot_isolation** | スナップショットアイソレーションが有効であるか無効であるかを示す値を返します。「allow_snapshot_isolation オプション」590 ページを参照してください。 |
| **allow_write_client_file** | データベースサーバーがクライアントコンピューターへのファイルの書き込みを許可するかどうかを示す値を返します。「allow_write_client_file オプション」592 ページを参照してください。 |
| **ansi_blanks** | どのような場合に文字データがクライアント側でトランケートされるかを示す値を返します。「ansi_blanks オプション」592 ページを参照してください。 |
| **ansi_close_cursors_on_rollback** | WITH HOLD で開かれたカーソルが ROLLBACK の実行時に閉じるかどうかを示す値を返します。「ansi_close_cursors_on_rollback オプション」593 ページを参照してください。 |
| **ansi_permissions** | DELETE 文と UPDATE 文についてパーミッションをチェックするかどうかを示す値を返します。「ansi_permissions オプション」594 ページを参照してください。 |
| **ansi_substring** | start パラメーターまたは length パラメーターに負の値が設定された場合に SUBSTRING (SUBSTR) 関数がどのような動作をするかを示す値を返します。「ansi_substring オプション」595 ページを参照してください。 |
| **ansi_update_constraints** | 更新可能な範囲を示す値を返します。「ansi_update_constraints オプション」596 ページを参照してください。 |
| **ansinull** | NULL をどのように解釈するかを示す値を返します。「ansinull オプション」597 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **AppInfo** | 接続を確立したクライアントに関する情報を返します。HTTP 接続では、ブラウザーの情報が含まれています。jConnect または Sybase Open Client の古いバージョンを使った接続については、情報は不完全の場合があります。<br><br>API 値は、DBLIB、ODBC、OLEDB、ADO.NET、iAnywhereJDBC、PHP、PerlDBD、DBEXPRESS のいずれかです。<br><br>その他のタイプの接続について返される値の詳細については、「AppInfo (APP) 接続パラメーター」321 ページを参照してください。 |
| **ApproximateCPUTime** | 特定の接続で使用された累積 CPU 時間の概算値 (秒単位) を返します。この値と実際の CPU 時間との誤差は、通常は 5 ～ 10 % ですが、最大で 50 % に達することもあります。マルチプロセッサーコンピューターでは、CPU (ハイパースレッドまたはコア) ごとに使用時間が積算されるため、すべての接続の累積時間を合計した値が実際の経過時間より大きくなる場合もあります。このプロパティは Windows と Linux でサポートされています。 |
| **auditing** | PUBLIC.auditing オプションが On に設定されている場合は、On を返します。それ以外の場合は、Off を返します。<br><br>PUBLIC.auditing オプションが On に設定され、conn_auditing オプションが Off に設定されている場合は、現在の接続が監査されていなくても、auditing 接続プロパティは On を返します。「監査の制御」1388 ページと「auditing オプション」598 ページを参照してください。 |
| **auditing_options** | このプロパティはシステムで使用するために予約されています。このオプションの設定は変更しないでください。 |
| **Authenticated** | アプリケーションが有効な接続認証文字列を送信した場合は Yes を返します。アプリケーションが有効な接続認証文字列を送信しなかった場合は No を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **AuthType** | 接続時に使用される認証のタイプを返します。返される値は、Standard、Integrated、Kerberos、空の文字列のいずれかです。空の文字列は、内部接続と AUTHORIZATION OFF を使用する HTTP サービスの接続の場合に返されます。 |
| **background_priority** | このプロパティは使用されなくなりました。現在の接続が他の接続のパフォーマンスにもたらす影響の大きさを示す値を返します。「background_priority オプション [廃止予定]」599 ページを参照してください。 |
| **BlockedOn** | 現在の接続がブロックされていない場合は 0 を返し、ブロックされている場合はロック競合によってブロックされた接続の接続番号を返します。 |
| **blocking** | ロック競合に対するデータベースサーバーの動作を示す値を返します。「blocking オプション」601 ページを参照してください。 |
| **blocking_others_timeout** | 現在の接続がロールバックされる前に、別の接続が現在の接続のローおよびテーブルロックでブロックされる可能性がある時間の長さを返します。「blocking_others_timeout オプション」601 ページを参照してください。 |
| **blocking_timeout** | トランザクションがロックを獲得するまでの待ち時間 (ミリ秒単位) を返します。「blocking_timeout オプション」602 ページを参照してください。 |
| **BytesReceived** | クライアント／サーバー通信中に受信したバイト数を返します。この値は、HTTP 接続と HTTPS 接続の場合に更新されます。 |
| **BytesReceivedUncomp** | 圧縮が無効になっている場合にクライアント／サーバー通信中に受信したバイト数を返します (この値は、圧縮が無効の場合は BytesReceived の値と同じ)。 |
| **BytesSent** | クライアント／サーバー通信中に送信されたバイト数を返します。この値は、HTTP 接続と HTTPS 接続の場合に更新されます。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **BytesSentUncomp** | 圧縮が無効になっている場合にクライアント／サーバー通信中に送信されたバイト数を返します (この値は、圧縮が無効の場合は BytesSent の値と同じ)。 |
| **CacheHits** | 成功したキャッシュ読み込み回数を返します。 |
| **CacheRead** | キャッシュ内で検索されたデータベースページの数を返します。 |
| **CacheReadIndInt** | キャッシュから読み込まれたインデックス内部ノードページの数を返します。 |
| **CacheReadIndLeaf** | キャッシュから読み込まれたインデックスリーフページの数を返します。 |
| **CacheReadTable** | キャッシュから読み込まれたテーブルページの数を返します。 |
| **CacheReadWorkTable** | キャッシュワークテーブルの読み込み数を返します。 |
| **CarverHeapPages** | クエリ最適化などの短期的な目的に使用されたヒープページの数を返します。 |
| **chained** | BEGIN TRANSACTION 文が指定されていない場合に使用されるトランザクションモードを返します。「chained オプション」603 ページを参照してください。 |
| **CharSet** | 接続で使用される CHAR 文字セットを返します。このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。「CONNECTION_EXTENDED_PROPERTY 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **checkpoint_time** | データベースサーバーが、あるチェックポイントを実行してから次のチェックポイントを実行するまでの最大時間 (分単位) を返します。「checkpoint_time オプション」603 ページを参照してください。 |

| プロパティ名 | 説明 |
| --- | --- |
| **cis_option** | リモートデータアクセス用のデバッグ情報がデータベースサーバーメッセージウィンドウに表示される場合は 7 を返し、表示されない場合は 0 を返します。「cis_option オプション」604 ページを参照してください。 |
| **cis_rowset_size** | 各フェッチにおいてリモートサーバーから返されるローの数を返します。「cis_rowset_size オプション」605 ページを参照してください。 |
| **ClientLibrary** | jConnect 接続の場合は jConnect、Sybase Open Client 接続の場合は CT_Library、HTTP 接続の場合は None、ODBC 接続、Embedded SQL 接続、OLE DB 接続、ADO.NET 接続、SQL Anywhere JDBC ドライバー接続の場合は CmdSeq を返します。 |
| **ClientNodeAddress** | クライアント／サーバー接続のクライアント側に対応するノードを返します。クライアントとサーバーの両方が同じコンピューターにある場合は、空の文字列を返します。これは NodeAddress プロパティの同意語です。<br><br>現在実行中の要求がイベントハンドラーの一部である場合、このプロパティは **NA** を返します。 |
| **ClientPort** | クライアントの TCP/IP ポート番号を返します。接続の種類が TCP/IP でない場合は、0 を返します。 |
| **ClientStmtCacheHits** | クライアントに文がキャッシュされていることが理由で、この接続には不要な準備文の数を返します。これは、クライアントでの文のキャッシュが無効になった場合に必要となる、追加の準備文の数です。<br>「max_client_statements_cached オプション」640 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **ClientStmtCacheMisses** | この接続において、クライアントのキャッシュに存在した文の中で、再び準備された文の数を返します。これは、キャッシュされた文が再使用できるか検討されたものの、スキーマの変更、データベースのオプション設定の変更、DROP VARIABLE 文によって再使用できなかった回数です。「max_client_statements_cached オプション」640 ページを参照してください。 |
| **close_on_endtrans** | トランザクションの終了時にカーソルを閉じるかどうかを示す値として、On または Off を返します。「close_on_endtrans オプション」605 ページを参照してください。 |
| **collect_statistics_on_dml_updates** | INSERT、DELETE、UPDATE などのデータを変更する DML 文の実行中に統計情報を収集するかどうかを示す値として、On または Off を返します。「collect_statistics_on_dml_updates オプション」606 ページを参照してください。 |
| **Commit** | 処理されたコミット要求の数を返します。 |
| **CommLink** | 接続の通信リンクを返します。これは SQL Anywhere がサポートするネットワークプロトコルであり、同一コンピューター接続の場合は local となります。<br><br>現在実行中の要求がイベントハンドラーの一部である場合、このプロパティは **NA** を返します。 |
| **CommNetworkLink** | 接続の通信リンクを返します。これは SQL Anywhere がサポートするネットワークプロトコルです。値には SharedMemory と TCPIP があります。CommNetworkLink プロパティは、同一コンピューターかどうかにかかわらず、常にリンク名を返します。<br><br>現在実行中の要求がイベントハンドラーの一部である場合、このプロパティは **NA** を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **CommProtocol** | Sybase Open Client 接続と jConnect 接続の場合は TDS、HTTP 接続の場合は HTTP、HTTPS 接続の場合は HTTPS、ODBC 接続、Embedded SQL 接続、OLE DB 接続、ADO.NET 接続、SQL Anywhere JDBC ドライバー接続の場合は CmdSeq を返します。 |
| **Compression** | この接続で通信圧縮が有効であるかどうかを示す値として、On または Off を返します。<br><br>現在実行中の要求がイベントハンドラーの一部である場合、このプロパティは **NA** を返します。 |
| **conn_auditing** | この接続に対して監査が有効になっている場合は、auditing オプションが Off に設定されていても、On を返します。「監査の制御」1388 ページと「conn_auditing オプション」607 ページを参照してください。 |
| **ConnectedTime** | 接続がこれまで接続してきた合計の時間数を秒単位で返します。 |
| **connection_authentication** | クライアントの認証に使用される文字列を返します。データベースが変更可能になる前に、認証が必要です。「connection_authentication オプション」608 ページを参照してください。 |
| **continue_after_raiserror** | RAISERROR 文が見つかるたびにプロシージャーまたはトリガーの実行を停止するかどうかを示す値として、On または Off を返します。「continue_after_raiserror オプション」609 ページを参照してください。 |
| **conversion_error** | データベースから情報をフェッチするときにデータ型変換障害をレポートするかどうかを示す値として、On または Off を返します。「conversion_error オプション」610 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
|---|---|
| **cooperative_commit_timeout** | データベースサーバーがディスクへの書き込み前に他の接続の書き込みによって 1 つのログページが満杯になるまで待っている時間 (ミリ秒単位) を返します。「cooperative_commit_timeout オプション」610 ページを参照してください。 |
| **cooperative_commits** | コミットをいつディスクに書き込むかを示す値として、On または Off を返します。「cooperative_commits オプション」611 ページを参照してください。 |
| **CurrentLineNumber** | 接続で実行されているプロシージャーまたは複合文の現在の行番号を返します。実行されているプロシージャーの名前は CurrentProcedure プロパティで取得できます。現在の行がクライアント側からの複合文の一部である場合は、空の文字列を返します。 |
| **CurrentProcedure** | 接続で現在実行されているプロシージャーの名前を返します。接続でネストされたプロシージャーコールが実行されている場合は、現在のプロシージャーの名前を返します。実行されているプロシージャーがない場合は、空の文字列を返します。 |
| **Cursor** | 現在データベースサーバーが保持している宣言されたカーソルの数を返します。 |
| **CursorOpen** | 現在データベースサーバーが保持している開いたカーソルの数を返します。 |
| **database_authentication** | データベースの認証に使用される文字列を返します。データベースサーバーの認証が完了してからでなければ、データベースに変更を加えることはできません。「database_authentication」612 ページを参照してください。 |
| **date_format** | データベースから取り出した日付のフォーマットを示す文字列を返します。「date_format オプション」613 ページを参照してください。 |
| **date_order** | 日付のフォーマット方法を示す文字列を返します。「date_order オプション」615 ページを参照してください。 |

| プロパティ名 | 説明 |
| --- | --- |
| **DBNumber** | データベースの ID 番号を返します。 |
| **debug_messages** | DEBUG ONLY 句を含む MESSAGE 文を実行するかどうかを示す値として、On または Off を返します。「debug_messages オプション」616 ページを参照してください。 |
| **dedicated_task** | 要求処理タスクがこの接続からの要求の処理専用であるかどうかを示す値として、On または Off を返します。「dedicated_task オプション」617 ページを参照してください。 |
| **default_dbspace** | デフォルト DB 領域の名前を返します。デフォルト DB 領域が指定されていない場合は、空の文字列を返します。「default_dbspace オプション」617 ページを参照してください。 |
| **default_timestamp_increment** | TIMESTAMP 型のカラム内の値を一意に保つために加算する値 (ミリ秒単位) を返します。「default_timestamp_increment オプション」618 ページを参照してください。 |
| **delayed_commit_timeout** | COMMIT の後でデータベースサーバーがアプリケーションに制御を戻すまでの待ち時間 (ミリ秒単位) を返します。「delayed_commit_timeout オプション」619 ページを参照してください。 |
| **delayed_commits** | COMMIT の後でデータベースサーバーがアプリケーションに制御を戻す時点を示す値として、On または Off を返します。「delayed_commits オプション」620 ページを参照してください。 |
| **DiskRead** | ディスクから読み込まれたページの数を返します。 |
| **DiskReadHint** | ディスクの読み込みヒント数を返します。 |
| **DiskReadHintPages** | ディスクの読み込みヒントページの数を返します。 |
| **DiskReadIndInt** | ディスクから読み込まれたインデックス内部ノードページの数を返します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
|---|---|
| **DiskReadIndLeaf** | ディスクから読み込まれたインデックスリーフページの数を返します。 |
| **DiskReadTable** | ディスクから読み込まれたテーブルページの数を返します。 |
| **DiskReadWorkTable** | ディスクワークテーブルの読み込み数を返します。 |
| **DiskSyncRead** | 同期的に発行されたディスクの読み込み数を返します。 |
| **DiskSyncWrite** | 同期的に発行された書き込み数を返します。 |
| **DiskWaitRead** | データベースサーバーによる非同期読み込みの待機回数を返します。 |
| **DiskWaitWrite** | データベースサーバーによる非同期書き込みの待機回数を返します。 |
| **DiskWrite** | ディスクに書き込まれた修正ページの数を返します。 |
| **DiskWriteHint** | ディスクの書き込みヒント数を返します。 |
| **DiskWriteHintPages** | ディスクの書き込みヒントページの数を返します。 |
| **divide_by_zero_error** | ゼロ除算によってエラーが発生する場合は On を返し、ゼロ除算によってエラーが発生しない場合は Off を返します。<br>「divide_by_zero_error オプション」622 ページを参照してください。 |
| **Encryption** | 接続が暗号化されるかどうかを示す値を返します。「Encryption (ENC) 接続パラメーター」342 ページを参照してください。 |
| **escape_character** | このプロパティはシステムで使用するために予約されています。このオプションの設定は変更しないでください。 |
| **EventName** | 接続でイベントハンドラーが実行されている場合、関連付けられているイベントの名前を返します。それ以外の場合は、空の文字列が返されます。 |

| プロパティ名 | 説明 |
| --- | --- |
| **exclude_operators** | このプロパティはシステムで使用するために予約されています。このオプションの設定は変更しないでください。 |
| **ExprCacheAbandons** | ヒット率が低すぎるために式キャッシュが中止された回数を返します。 |
| **ExprCacheDropsToReadOnly** | ヒット率が低いために式キャッシュが読み込み専用ステータスに変更された回数を返します。 |
| **ExprCacheEvicts** | 式キャッシュからの出力数を返します。 |
| **ExprCacheHits** | 式キャッシュ内のヒット数を返します。 |
| **ExprCacheInserts** | 式キャッシュに挿入された値の数を返します。 |
| **ExprCacheLookups** | 式キャッシュ内で実行されたルックアップの回数を返します。 |
| **ExprCacheResumesOfReadWrite** | ヒット率の上昇によって式キャッシュが読み込み／書き込みステータスに戻った回数を返します。 |
| **ExprCacheStarts** | 式キャッシュが開始された回数を返します。 |
| **extended_join_syntax** | 複数テーブルのジョインに対して、重複する相関名構文を持つクエリが許可されている場合は On、エラーとしてレポートされる場合は Off を返します。「extended_join_syntax オプション」623 ページを参照してください。 |
| **fire_triggers** | データベースにおいてトリガーが起動する場合は On を返し、起動しない場合は Off を返します。「fire_triggers オプション」624 ページを参照してください。 |
| **first_day_of_week** | 週の最初の曜日を表す番号を返します。7 は日曜日を表し、1 は月曜日を表します。「first_day_of_week オプション」625 ページを参照してください。 |

| プロパティ名 | 説明 |
|---|---|
| **for_xml_null_treatment** | NULL 値を含む要素と属性が結果から除外される場合は OMIT を返し、クエリ内で FOR XML 句が使用されている場合に NULL 値に対応する空の要素または属性が生成される場合は EMPTY を返します。<br>「for_xml_null_treatment オプション」625 ページを参照してください。 |
| **force_view_creation** | このプロパティはシステムで使用するために予約されています。このオプションの設定は変更しないでください。 |
| **FullCompare** | インデックスのハッシュ値を超えて実行された比較の回数を返します。 |
| **GetData** | GETDATA 要求の数を返します。 |
| **global_database_id** | DEFAULT GLOBAL AUTOINCREMENT が指定されたカラムの開始値を返します。<br>「global_database_id オプション」626 ページを参照してください。 |
| **HashForcedPartitions** | メモリ競合が原因でハッシュ演算子が分割された回数を返します。 |
| **HashRowsFiltered** | ビットベクトルフィルターによって拒否されたプローブロー数を返します。 |
| **HashRowsPartitioned** | ハッシュワークテーブルに書き込まれたロー数を返します。 |
| **HashWorkTables** | ハッシュベースの演算用に作成されたワークテーブルの数を返します。 |
| **HeapsCarver** | クエリ最適化などの短期的な目的に使用されたヒープの数を返します。 |
| **HeapsLocked** | キャッシュ内で現在ロックされている再配置可能なヒープの数を返します。 |
| **HeapsQuery** | クエリ処理 (ハッシュ操作とソート操作) に使用されるヒープの数を返します。 |
| **HeapsRelocatable** | 再配置可能なヒープの数を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **http_connection_pool_basesize** | データベース接続の公称スレッショルドサイズを返します。「http_connection_pool_basesize オプション」627 ページを参照してください。 |
| **http_connection_pool_timeout** | 未使用の接続が接続プールに格納される最長時間を返します。<br>「http_connection_pool_timeout オプション」628 ページを参照してください。 |
| **http_session_timeout** | 現在の HTTP セッションタイムアウトを返します (分単位)。「http_session_timeout オプション」629 ページを参照してください。 |
| **HttpServiceName** | 現在の HTTP 要求のサービス名エントリポイントを返します。このプロパティは、エラーレポートやフロー制御の場合に便利です。このプロパティを HTTP 要求以外により実行されたストアドプロシージャーから選択した場合、または接続が現在アクティブではないか、HTTP セッションの再開を待っている状態である場合は、空の文字列を返します。 |
| **IdleTimeout** | 接続のアイドルタイムアウト値を返します。「Idle 接続パラメーター」348 ページを参照してください。<br><br>現在実行中の要求がイベントハンドラーの一部である場合、このプロパティは **NA** を返します。 |
| **IndAdd** | インデックスに追加されたエントリの数を返します。 |
| **IndLookup** | インデックス内で検索されたエントリの数を返します。 |
| **integrated_server_name** | 統合化ログインを目的とする Windows ユーザーグループメンバーシップの検索に使用されるドメインコントローラサーバーの名前を返します。「integrated_server_name オプション」630 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **IsDebugger** | SQL Anywhere デバッガーの実行に使用されている接続を区別するために Yes または No を返します。現在の接続番号がデバッガー接続の接続番号と対応する場合は値が Yes、それ以外の場合は No になります。「SQL Anywhere のデバッガー」『SQL Anywhere サーバー SQL の使用法』を参照してください。 |
| **isolation_level** | 接続の独立性レベルを返します (0、1、2、3、Snapshot、Statement-snapshot、または Readonly-statement-snapshot)。「isolation_level オプション」630 ページを参照してください。 |
| **java_class_path** | クラスを検索する追加のディレクトリまたは JAR ファイルのリストを返します。「java_class_path option」632 ページを参照してください。 |
| **java_location** | データベースに対して Java VM が指定されている場合、そのパスを返します。「java_location オプション」633 ページを参照してください。 |
| **java_main_userid** | このプロパティは使用されなくなりました。 |
| **java_vm_options** | Java VM の起動時にデータベースサーバーが使用するコマンドラインオプションを返します。「java_vm_options オプション」633 ページを参照してください。 |
| **Language** | ロケール言語を返します。 |
| **LastCommitRedoPos** | 接続によって最後の COMMIT 操作がトランザクションログに書き込まれた後で REDO ログの位置を返します。 |
| **LastIdle** | 要求間のチック数を返します。 |
| **LastPlanText** | 接続で最後に実行されたクエリの長いテキストプランを返します。最後のプランを記憶するかどうかは、sa_server_option システムプロシージャーの RememberLastPlan オプションを設定するか、-zp サーバーオプションを使用することで制御します。「-zp dbeng12/dbsrv12 サーバーオプション」301 ページを参照してください。 |

| プロパティ名 | 説明 |
| --- | --- |
| **LastReqTime** | 指定された接続において最後の要求が開始された時刻を返します。このプロパティは、イベントなどの内部接続の場合は空の文字列を返すことがあります。 |
| **LastStatement** | 現在の接続で最後に準備された SQL 文を返します。<br><br>LastStatement の値は、文が準備されると同時に設定され、文が削除されると同時にクリアされます。各接続につき 1 つの文の文字列のみが記憶されます。<br><br>ある接続について sa_conn_activity が空でない値を返した場合、その接続で現在実行されている文である可能性が高くなります。その文が完了している場合は、文がすでに削除され、このプロパティの値がクリアされている可能性があります。アプリケーションが複数の文を準備し、それらの文のステートメントハンドルを保持している場合、LastStatement が返す値は接続で現在実行されている処理を表しません。<br><br>クライアントでの文のキャッシュが有効であり、キャッシュされた文が再使用されているとき、このプロパティは空の文字列を返します。 |
| **LivenessTimeout** | 現在の接続の活性タイムアウト時間を返します。「LivenessTimeout (LTO) 接続パラメーター」352 ページを参照してください。<br><br>現在実行中の要求がイベントハンドラーの一部である場合、このプロパティは **NA** を返します。 |
| **lock_rejected_rows** | このプロパティはシステムで使用するために予約されています。このオプションの設定は変更しないでください。 |
| **LockCount** | 接続で保持されているロックの数を返します。 |
| **LockIndexID** | ロックされたインデックスの識別子を返します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
|---|---|
| **LockName** | 接続が解放を待っているロックを示す 64 ビット符号なし整数を返します。 |
| **LockRowID** | ロックされたローの識別子を返します。 |
| **LockTableOID** | 接続がブロックされていない場合、または接続が CONNECTION_PROPERTY を呼び出した接続とは別のデータベース上にある場合は、0 を返します。それ以外の場合は、接続が解放を待っているロックが設定されているテーブルのオブジェクト ID を返します。このオブジェクト ID によって、SYSTAB システムビューを使用してテーブル情報を検索できます。「SYSTAB システムビュー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **log_deadlocks** | デッドロック情報がレポートされる場合は On を返し、それ以外の場合は Off を返します。「log_deadlocks オプション」634 ページを参照してください。 |
| **LogFreeCommit** | Redo Free Commit の数を返します。Redo Free Commit が発生するのは、トランザクションログのコミットが要求されているが、ログはすでに書き込まれている (コミットが実行されている) 場合です。 |
| **login_mode** | サポートされているログインのタイプを示す値として、Standard、Integrated、CloudAdmin、Kerberos の中から 1 つまたは複数を返します。「login_mode オプション」635 ページを参照してください。 |
| **login_procedure** | 起動時に互換性オプションを設定するストアドプロシージャーの名前を返します。「login_procedure オプション」636 ページを参照してください。 |
| **LoginTime** | 接続が確立された日付と時刻を返します。 |
| **LogWrite** | トランザクションログに書き込まれたページの数を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **materialized_view_optimization** | マテリアライズドビューをクエリ最適化の実行中に使用するかどうかを示す値を返します。<br><br>● Disabled<br>● Fresh<br>● Stale<br>● *N* Minute[s]<br>● *N* Hour[s]<br>● *N* Day[s]<br>● *N* Week[s]<br>● *N* Month[s]<br><br>「materialized_view_optimization オプション」639 ページを参照してください。 |
| **max_client_statements_cached** | クライアントでキャッシュされる文の数を返します。「max_client_statements_cached オプション」640 ページを参照してください。 |
| **max_cursor_count** | 接続で一度に使用できるカーソルの最大数を返します。「max_cursor_count オプション」642 ページを参照してください。 |
| **max_hash_size** | このプロパティは使用されなくなりました。 |
| **max_plans_cached** | キャッシュに格納される実行プランの最大数を返します。「max_plans_cached オプション」643 ページを参照してください。 |
| **max_priority** | 接続に設定できる最高優先度レベルを示す値を返します。「max_priority オプション」643 ページを参照してください。 |
| **max_query_tasks** | データベースサーバーがクエリの処理に使用できる要求の最大数を返します。<br>「max_query_tasks オプション」644 ページを参照してください。 |
| **max_recursive_iterations** | 再帰共通テーブル式が実行できる反復処理の最大回数を返します。<br>「max_recursive_iterations オプション」645 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **max_statement_count** | 接続で一度に使用できる準備文の最大数を返します。「max_statement_count オプション」646 ページを参照してください。 |
| **max_temp_space** | 1 つの接続で使用できるテンポラリファイル領域の最大サイズを返します。「max_temp_space オプション」647 ページを参照してください。 |
| **MessageReceived** | MESSAGE 文によって生成され、WAITFOR 文が中断する原因となった文字列を返します。それ以外の場合は、空の文字列が返されます。 |
| **min_password_length** | データベースにおける新しいパスワードの最小長を返します。「min_password_length オプション」648 ページを参照してください。 |

| プロパティ名 | 説明 |
| --- | --- |
| **Name** | 現在の接続の名前を返します。<br><br>ConnectionName (CON) 接続パラメーターを使用して接続名を指定できます。<br>「ConnectionName (CON) 接続パラメーター」332 ページを参照してください。<br><br>データベースサーバーによって作成されるテンポラリ接続には次の名前が使用されます。<br><br>● INT:ApplyRecovery<br>● INT:BackupDB<br>● INT:Checkpoint<br>● INT:Cleaner<br>● INT:CloseDB<br>● INT:CreateDB<br>● INT:CreateMirror<br>● INT:DelayedCommit<br>● INT:DiagRcvr<br>● INT:DropDB<br>● INT:EncryptDB<br>● INT:Exchange<br>● INT:FlushMirrorLog<br>● INT:FlushStats<br>● INT:HTTPReq<br>● INT:PromoteMirror<br>● INT:PurgeSnapshot<br>● INT:ReconnectMirror<br>● INT:RecoverMirror<br>● INT:RedoCheckpoint<br>● INT:RefreshIndex<br>● INT:ReloadTrigger<br>● INT:RenameMirror<br>● INT:RestoreDB<br>● INT:StartDB<br>● INT:VSS<br><br>「テンポラリ接続」76 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **NcharCharSet** | 接続で使用される NCHAR 文字セットを返します。このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。「CONNECTION_EXTENDED_PROPERTY 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **nearest_century** | 文字列から日付への変換において 2 桁の年がどのように解釈されるかを示す値を返します。「nearest_century オプション」649 ページを参照してください。 |
| **NodeAddress** | クライアント／サーバー接続のクライアント側に対応するノードを返します。クライアントとサーバーの両方が同じコンピューターにある場合は、空の文字列を返します。 |
| **non_keywords** | オフになっているため識別子として使用できるキーワードのリストを返す (そのようなキーワードが存在する場合)。「non_keywords オプション」650 ページを参照してください。 |
| **Number** | 接続の ID 番号を返します。 |
| **odbc_describe_binary_as_varbinary** | SQL Anywhere ODBC ドライバーが BINARY カラムと VARBINARY カラムをどちらも SQL_BINARY として記述する場合は Off を返し、ODBC ドライバーが BINARY カラムと VARBINARY カラムを SQL_VARBINARY として記述する場合は On を返します。「odbc_describe_binary_as_varbinary」650 ページを参照してください。 |
| **odbc_distinguish_char_and_varchar** | CHAR カラムが SQL_VARCHAR として記述される場合は Off を返し、SQL_CHAR として記述される場合は On を返します。「odbc_distinguish_char_and_varchar オプション」651 ページを参照してください。 |
| **oem_string** | データベースファイルのヘッダーページ内の文字列を返します。「oem_string オプション」652 ページを参照してください。 |

| プロパティ名 | 説明 |
| --- | --- |
| **on_charset_conversion_failure** | 文字セットの変換中にエラーが発生した場合の動作を示す値として、Ignore、Warning、Error のいずれかを返します。<br>「on_charset_conversion_failure オプション」654 ページを参照してください。 |
| **on_tsql_error** | ストアドプロシージャーまたは T-SQL バッチの実行中にエラーが発生した場合の動作を示す値として、Stop、Conditional、Continue のいずれかを返します。「on_tsql_error オプション」655 ページを参照してください。 |
| **optimization_goal** | クエリ処理を最適化する方法を示す値として、First-row または All-rows を返します。<br>「optimization_goal オプション」655 ページを参照してください。 |
| **optimization_level** | 0 ～ 15 の範囲の値を返します。この値を使用して、SQL Anywhere クエリオプティマイザーが SQL 文のアクセスプランの検索に費やす作業レベルを指定します。<br>「optimization_level オプション」657 ページを参照してください。 |
| **optimization_workload** | SQL Anywhere クエリオプティマイザーが SQL 文のアクセスプランの検索に費やす作業レベルを示す値を返します。<br>「optimization_workload オプション」658 ページを参照してください。 |
| **OSUser** | クライアントプロセスに関連付けられたオペレーティングシステムユーザー名を返します。クライアントプロセスで別のユーザーを同一化している場合 (UNIX の場合は、セット ID ビットが設定されている場合)、同一化されたユーザー名が返されます。バージョン 10.0.1 以前のクライアントと HTTP や TDS クライアントでは、空の文字列を返します。 |
| **PacketSize** | 接続で使用されるパケットサイズ (バイト単位) を返します。<br><br>現在実行中の要求がイベントハンドラーの一部である場合、このプロパティは **NA** を返します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **PacketsReceived** | 受信したクライアント／サーバー通信パケットの数を返します。この値は、HTTP 接続と HTTPS 接続の場合は更新されません。 |
| **PacketsReceivedUncomp** | 圧縮が無効になっている場合にクライアント／サーバー通信中に受信したパケットの数を返します (この値は、圧縮が無効の場合は PacketsReceived の値と同じ)。 |
| **PacketsSent** | 送信されたクライアント／サーバー通信パケットの数を返します。この値は、HTTP 接続と HTTPS 接続の場合は更新されません。 |
| **PacketsSentUncomp** | 圧縮が無効になっている場合にクライアント／サーバー通信中に送信されたパケットの数を返します (この値は、圧縮が無効の場合は PacketsSent の値と同じ)。 |
| **ParentConnection** | データベースオペレーション (データベースのバックアップや作成など) を実行するためにテンポラリ接続を作成した接続の接続 ID を返します。その他のタイプの接続について、このプロパティは NULL を返します。 |
| **pinned_cursor_percent_of_cache** | カーソルを固定するために使用できるキャッシュの割合を返します。<br>「pinned_cursor_percent_of_cache オプション」659 ページを参照してください。 |
| **post_login_procedure** | ユーザーが接続したときにアプリケーションが表示する必要のあるメッセージがプロシージャーの結果セットに含まれている場合、そのプロシージャーの名前を返します。<br>「post_login_procedure オプション」659 ページを参照してください。 |
| **precision** | 10 進数精度と数値精度の設定を返します。<br>「precision オプション」661 ページを参照してください。 |

| プロパティ名 | 説明 |
| --- | --- |
| **prefetch** | プリフェッチが行われない場合は Off、カーソルタイプが SENSITIVE であるかクエリにプロキシテーブルが含まれていないかぎりプリフェッチが行われる場合は Conditional、カーソルタイプが SENSITIVE であり、カーソルにプロキシテーブルが含まれていても常にプリフェッチが行われる場合は Always を返します。「prefetch オプション」662 ページを参照してください。 |
| **Prepares** | 接続に対して実行された文の準備作業の数を返します。 |
| **PrepStmt** | 現在データベースサーバーが保持している準備文の数を返します。 |
| **preserve_source_format** | プロシージャー、トリガー、ビュー、イベントハンドラーの元のソース定義がシステムファイルに保存される場合は On を返し、それ以外の場合は Off を返します。「preserve_source_format オプション」663 ページを参照してください。 |
| **prevent_article_pkey_update** | パブリケーションで使用されるテーブルのプライマリキーカラムの更新が許可されている場合は On を返し、それ以外の場合は Off を返します。「prevent_article_pkey_update オプション」664 ページを参照してください。 |
| **priority** | 接続の優先度レベルを示す値を返します。「priority オプション」664 ページを参照してください。 |
| **Progress** | クエリが実行されている時間に関する情報を返します。次に例を示します。<br><br>43% (9728 of 22230 pages) complete after 00:00:05; estimated 00:00:06 remaining<br><br>このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。「CONNECTION_EXTENDED_PROPERTY 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **progress_messages** | progress_messages オプションの値を返します。「progress_messages オプション」665 ページを参照してください。 |
| **query_mem_timeout** | query_mem_timeout オプションの値を返します。「query_mem_timeout オプション」667 ページを参照してください。 |
| **QueryBypassed** | オプティマイザーバイパスによって最適化された要求の数を返します。 |
| **QueryBypassedCosted** | オプティマイザーバイパスによってコストを使用して処理された要求の数を返します。 |
| **QueryBypassedHeuristic** | オプティマイザーバイパスによってヒューリスティックを使用して処理された要求の数を返します。 |
| **QueryBypassedOptimized** | オプティマイザーバイパスによって最初に処理されてから、SQL Anywhere オプティマイザーによって完全に最適化された要求の数を返します。 |
| **QueryCachedPlans** | 接続において現在キャッシュされているクエリ実行プランの数を返します。 |
| **QueryCachePages** | 実行プランの保存に使用されるキャッシュページの数を返します。 |
| **QueryDescribedBypass** | オプティマイザーバイパスによって処理された DESCRIBE 要求の数を返します。 |
| **QueryDescribedOptimizer** | オプティマイザーによって処理された DESCRIBE 要求の数を返します。 |
| **QueryHeapPages** | クエリ処理 (ハッシュ操作とソート操作) に使用されるキャッシュページの数を返します。 |
| **QueryJHToJNLOptUsed** | ハッシュジョインがネストループジョインに変換された回数を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **QueryLowMemoryStrategy** | メモリ不足が原因でサーバーが実行プランを実行中に変更した回数を返します。現在使用できるメモリがオプティマイザーの推定よりも少ない、または実行プランが必要とするメモリがオプティマイザーの推定よりも多い場合に、プランが変更されることがあります。 |
| **QueryMemActiveCurr** | クエリメモリをアクティブに使用する要求数を返します。 |
| **QueryMemGrantFailed** | 要求がクエリメモリ待ちになり、取得できなかった合計回数を返します。 |
| **QueryMemGrantGranted** | 要求に現在付与されているページ数を返します。 |
| **QueryMemGrantRequested** | 要求がクエリメモリを取得しようとした合計回数を返します。 |
| **QueryMemGrantWaited** | 要求がクエリメモリ待ちになった合計回数を返します。 |
| **QueryMemGrantWaiting** | クエリメモリ待ちになっている現在の要求数を返します。 |
| **QueryOpened** | 実行対象の OPEN 要求の数を返します。 |
| **QueryOptimized** | 完全に最適化された要求の数を返します。 |
| **QueryReused** | プランキャッシュから再利用された要求の数を返します。 |
| **QueryRowsFetched** | この接続について、逐次スキャンまたはインデックススキャンによってベーステーブルから読み込まれたローの数を返します。 |
| **QueryRowsMaterialized** | クエリ処理中にワークテーブルに書き込まれるロー数を返します。 |
| **quoted_identifier** | 二重引用符で囲まれた文字列が識別子として解釈される場合は On を返し、リテラル文字列として解釈される場合は Off を返します。「quoted_identifier オプション」668 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
|---|---|
| **read_past_deleted** | 独立性レベル 1 または 2 の逐次スキャンにおいてコミットされていない削除ローが無視される場合は On を返し、独立性レベル 1 または 2 の逐次スキャンがコミットされていない削除ローでブロックされる場合は Off を返します。「read_past_deleted オプション」669 ページを参照してください。 |
| **recovery_time** | データベースサーバーがシステム障害から回復するのに要する最長時間 (分単位) を返します。「recovery_time オプション」669 ページを参照してください。 |
| **RecursiveIterations** | 再帰ユニオンの反復回数を返します。 |
| **RecursiveIterationsHash** | 再帰ハッシュジョインでハッシュ方式が使用された回数を返します。 |
| **RecursiveIterationsNested** | 再帰ハッシュジョインでネストループ方式が使用された回数を返します。 |
| **RecursiveJNLMisses** | 再帰ハッシュジョインでのインデックスプローブキャッシュミスの数を返します。 |
| **RecursiveJNLProbes** | 再帰ハッシュジョインにおけるインデックスプローブの試行回数を返します。 |
| **remote_idle_timeout** | Web サービスのクライアントプロシージャーと関数で許容される休止時間 (秒単位) を返します。「remote_idle_timeout オプション」670 ページを参照してください。 |
| **replicate_all** | 内部でのみ使用されます。 |
| **ReqCountActive** | 処理が完了した要求の数を返します。RequestTiming サーバープロパティが Off に設定されている場合は、NULL を返します。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |
| **ReqCountBlockContention** | 接続がアトミックアクセスを待った回数を返します。-zt オプションが指定されていない場合は、NULL を返します。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |

| プロパティ名 | 説明 |
| --- | --- |
| **ReqCountBlockIO** | 接続が I/O 処理の完了を待った回数を返します。-zt オプションが指定されていない場合は、NULL を返します。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |
| **ReqCountBlockLock** | 接続がロックの解放を待った回数を返します。-zt オプションが指定されていない場合は、NULL を返します。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |
| **ReqCountUnscheduled** | 接続がスケジューリングを待った回数を返します。-zt オプションが指定されていない場合は、NULL を返します。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **ReqStatus** | 要求のステータスを返します。値は次のいずれかです。<br>● **Idle**　この接続では現在、要求を処理していない。<br>● **Unscheduled***　この接続では他の処理が実行されており、使用可能なデータベースサーバーワーカーを待っている。<br>● **BlockedIO***　この接続はブロックされ、I/O 処理の完了を待っている。<br>● **BlockedContention***　この接続はブロックされ、共有データベースサーバーデータ構造体へのアクセスを待っている。<br>● **BlockedLock**　この接続はブロックされ、オブジェクトのロックの解放を待っている。<br>● **Executing**　この接続では現在、要求を実行している。<br>アスタリスク (*) が付いている値が返されるのは、-zt サーバーオプションによってデータベースサーバーで要求タイミング情報のロギングがオンになっている場合に限られます。要求タイミング情報のロギングが実行されていない場合は (デフォルト)、値は Executing として返されます。<br>「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |
| **ReqTimeActive** | 要求の処理に要した時間を返します。-zt オプションが指定されていない場合は、NULL を返します。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |
| **ReqTimeBlockContention** | アトミックアクセスを取得するまでの待ち時間を返します。RequestTiming サーバープロパティが Off に設定されている場合は、NULL を返します。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |

| プロパティ名 | 説明 |
| --- | --- |
| **ReqTimeBlockIO** | I/O 処理が完了するまでの待ち時間を返します。-zt オプションが指定されていない場合は、NULL を返します。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |
| **ReqTimeBlockLock** | ロックが解放されるまでの待ち時間を返します。-zt オプションが指定されていない場合は、NULL を返します。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |
| **ReqTimeUnscheduled** | 未スケジュール時間を返します。-zt オプションが指定されていない場合は、NULL を返します。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |
| **ReqType** | 最後の要求のタイプを返します。接続が接続プールによってキャッシュされる場合、その ReqType 値は CONNECT_POOL_CACHE になります。 |
| **request_timeout** | 1 つの要求を実行できる最大時間を返します。「request_timeout オプション」672 ページを参照してください。 |
| **RequestsReceived** | クライアント／サーバー通信要求またはラウンドトリップの数を返します。このプロパティでは、PacketsReceived とは異なり、マルチパケット要求を 1 つの要求として数え、活性パケットを計数の対象から除外します。 |
| **reserved_keywords** | データベースで有効なデフォルト以外の予約キーワードを返します。「reserved_keywords オプション」673 ページを参照してください。 |
| **return_date_time_as_string** | DATE、TIME、TIMESTAMP の値が文字列としてアプリケーションに返される場合は On を返し、DATE データ型または TIME データ型として返される場合は Off を返します。「return_date_time_as_string オプション」674 ページを参照してください。 |
| **Rlbk** | 処理されたロールバック要求の数を返します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **rollback_on_deadlock** | 参照整合性アクションが UPDATE または DELETE の後に実行される場合は After を返し、UPDATE または DELETE の前に実行される場合は Before を返します。「rollback_on_deadlock」674 ページを参照してください。 |
| **RollbackLogPages** | ロールバックログのページ数を返します。 |
| **row_counts** | ローの数が常に正確である場合は On を返し、ローの数が通常は推定値である場合は Off を返します。「row_counts オプション」675 ページを参照してください。 |
| **scale** | 接続における 10 進数と数値の位取りを返します。「scale オプション」676 ページを参照してください。 |
| **secure_feature_key** | データベースサーバーにおいて機能を有効または無効にするキーを格納します。このプロパティの値を選択すると、常に空の文字列が返されます。 |
| **ServerNodeAddress** | クライアント／サーバー接続のサーバー側に対応するノードを返します。クライアントとサーバーの両方が同じコンピューターにある場合は、空の文字列を返します。<br><br>現在実行中の要求がイベントハンドラーの一部である場合、このプロパティは **NA** を返します。 |
| **ServerPort** | データベースサーバーの TCP/IP ポート番号または 0 を返します。 |
| **SessionCreateTime** | HTTP セッションが作成された時刻を返します。 |
| **SessionID** | 接続のセッション ID がある場合は、そのセッション ID を返し、それ以外の場合は空の文字列を返します。 |
| **SessionLastTime** | HTTP セッションにおける最後の要求の時刻を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **SessionTimeout** | 非アクティブ状態の HTTP セッションが維持される時間 (分単位) を返します。<br>「sa_set_http_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **SnapshotCount** | 接続に関連付けられているスナップショットの数を返します。 |
| **sort_collation** | ORDER BY 句が変更されていない場合は Internal を返し、それ以外の場合は照合名または照合 ID を返します。「sort_collation オプション」678 ページを参照してください。 |
| **SortMergePasses** | ソート中に使用されたマージパスの数を返します。 |
| **SortRowsMaterialized** | ソートワークテーブルに書き込まれたロー数を返します。 |
| **SortRunsWritten** | ソート中に書き込まれたソート実行の数を返します。 |
| **SortSortedRuns** | 実行の生成中に作成されたソート実行の数を返します。 |
| **SortWorkTables** | ソート用に作成されたワークテーブルの数を返します。 |
| **sql_flagger_error_level** | 指定された SQL/2003 に含まれていない場合にエラーとして通知する SQL を示す値として、次のいずれかを返します。<br>● **E**　初級レベル SQL/2003 構文でない構文を通知します。<br>● **I**　中級レベル SQL/2003 構文でない構文を通知します。<br>● **F**　上級レベル SQL/2003 構文でない構文を通知します。<br>● **W**　サポートされている構文をすべて許可します。<br>「sql_flagger_error_level オプション」679 ページを参照してください。 |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **sql_flagger_warning_level** | 指定された SQL/2003 に含まれていない場合に警告として通知する SQL を示す値として、次のいずれかを返します。<br>● **E** 初級レベル SQL/2003 構文でない構文を通知します。<br>● **I** 中級レベル SQL/2003 構文でない構文を通知します。<br>● **F** 上級レベル SQL/2003 構文でない構文を通知します。<br>● **W** サポートされている構文をすべて許可します。<br>「sql_flagger_warning_level オプション」680 ページを参照してください。 |
| **st_geometry_asbinary_format** | 空間値がジオメトリからバイナリに変換される方法を示す値を返します。「st_geometry_asbinary_format オプション」686 ページを参照してください。 |
| **st_geometry_astext_format** | 空間値がジオメトリからテキストに変換される方法を示す値を返します。「st_geometry_astext_format オプション」687 ページを参照してください。 |
| **st_geometry_asxml_format** | 空間値がジオメトリから xml に変換される方法を示す値を返します。「st_geometry_asxml_format オプション」688 ページを参照してください。 |
| **st_geometry_describe_type** | 空間データ値を記述する方法を示す値を返します。「st_geometry_describe_type オプション」689 ページを参照してください。 |
| **st_geometry_interpolation** | ST_CircularString ジオメトリの内挿法設定を返します。「st_geometry_interpolation オプション」690 ページを参照してください。 |
| **st_geometry_on_invalid** | ジオメトリが表面検証に失敗した場合の動作を示す値を返します。「st_geometry_on_invalid オプション」691 ページを参照してください。 |

| プロパティ名 | 説明 |
| --- | --- |
| **StatementDescribes** | DESCRIBE 要求によって処理された文の合計数を返します。 |
| **StatementPostAnnotates** | セマンティックなクエリ変換フェーズによって処理された文の数を返します。 |
| **StatementPostAnnotatesSimple** | セマンティックなクエリ変換フェーズによって処理されたが、一部のセマンティック変換が省略された文の数を返します。 |
| **StatementPostAnnotatesSkipped** | セマンティックなクエリ変換フェーズが完全に省略された文の数を返します。 |
| **string_rtruncation** | 文字列がトランケートされたときにエラーが発生する場合は On を返し、エラーなしで文字列がトランケートされる場合は Off を返します。「string_rtruncation オプション」691 ページを参照してください。 |
| **subsume_row_locks** | データベースサーバーがテーブルに対する個別ローロックを取得する場合は On を返し、それ以外の場合は Off を返します。「subsume_row_locks オプション」693 ページを参照してください。 |
| **suppress_tds_debugging** | TDS デバッグ情報がデータベースサーバーメッセージウィンドウに表示される場合は Off を返し、表示されない場合は On を返します。「suppress_tds_debugging オプション」693 ページを参照してください。 |
| **synchronize_mirror_on_commit** | データベースミラーサーバーがコミット時に同期される場合は On を返し、それ以外の場合は Off を返します。「synchronize_mirror_on_commit オプション」694 ページを参照してください。 |
| **tds_empty_string_is_null** | TDS 接続において空の文字列が NULL として返される場合は On を返し、TDS 接続において 1 つのブランク文字を含む文字列が返される場合は Off を返します。「tds_empty_string_is_null オプション」694 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **temp_space_limit_check** | データベースサーバーが接続で使用できるテンポラリ領域のサイズをチェックする場合は On を返し、データベースサーバーが接続で使用できる領域のサイズをチェックしない場合は Off を返します。「temp_space_limit_check オプション」695 ページを参照してください。 |
| **TempFilePages** | 接続で使用されるテンポラリファイルページの数を返します。 |
| **TempTablePages** | テンポラリテーブルで使用されるテンポラリファイルのページ数を返します。 |
| **time_format** | データベースから取り出される時刻の文字列フォーマットを返します。「time_format オプション」696 ページを参照してください。 |
| **time_zone_adjustment** | 接続のローカル時間を表示するために協定世界時 (UTC: Coordinated Universal Time) に加算する必要がある時間 (分単位) を返します。「time_zone_adjustment オプション」697 ページを参照してください。 |
| **timestamp_format** | データベースから取り出したタイムスタンプのフォーマットを返します。「timestamp_format オプション」698 ページを参照してください。 |
| **timestamp_with_time_zone_format** | データベースから取り出した TIMESTAMP WITH TIME ZONE 値のフォーマットを返します。「timestamp_with_time_zone_format オプション」700 ページを参照してください。 |
| **TimeZoneAdjustment** | 接続のローカル時間を表示するために協定世界時 (UTC: Coordinated Universal Time) に加算する必要がある時間 (分単位) を返します。「time_zone_adjustment オプション」697 ページを参照してください。 |
| **TransactionStartTime** | COMMIT または ROLLBACK の後にデータベースが最初に変更された時刻を含む文字列を返します。最後の COMMIT または ROLLBAK 以降にデータベースが変更されていない場合は、空の文字列を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **truncate_timestamp_values** | TIMESTAMP 値で使用される小数点以下の桁数に制限がある場合は On を返し、ない場合は Off を返します。「truncate_timestamp_values オプション」702 ページを参照してください。 |
| **tsql_outer_joins** | DML 文で Transact-SQL 外部ジョインが使用できる場合は On を返します。「tsql_outer_joins オプション」703 ページを参照してください。 |
| **tsql_variables** | Embedded SQL のホスト変数名のプレフィクスとしてコロンの代わりに @ 符号を使用できる場合は On を返し、それ以外の場合は Off を返します。「tsql_variables オプション」703 ページを参照してください。 |
| **UncommitOp** | コミットされていない操作の数を返します。 |
| **updatable_statement_isolation** | isolation_level オプションが Readonly-statement-snapshot に設定されている場合に更新可能な文に使用される独立性レベル (0、1、2、または 3) を返します。「updatable_statement_isolation オプション」704 ページを参照してください。 |
| **update_statistics** | この接続が統計ガバナーにクエリフィードバックを送信できる場合、On を返します。update_statistics オプションが Off に設定されている場合、統計ガバナーは現在の接続からクエリフィードバックを受け取りません。「update_statistics オプション」705 ページを参照してください。 |
| **upgrade_database_capability** | このプロパティはシステムで使用するために予約されています。このオプションの設定は変更しないでください。 |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **user_estimates** | クエリオプティマイザーがクエリの述部に含まれる選択性推定を尊重するか無視するかを示す値として、次のいずれかを返します。<br>● **Enabled** ユーザーが提供する選択性推定をすべて尊重します。このオプションは、On を使用して有効にすることもできます。<br>● **Override-Magic** ユーザー選択性推定は尊重されますが、オプティマイザーが最後の手段であるヒューリスティック値 (マジック値とも呼ばれます) の使用を選択する以外に方法がない場合のみ使用されます。<br>● **Disabled** 他の推定データが使用できないときは、ユーザー推定は無視され、マジック値が使用されます。このオプションは、Off を使用して無効にすることもできます。<br>「user_estimates オプション」706 ページを参照してください。 |
| **UserAppInfo** | AppInfo 接続パラメーターによって接続文字列に指定された文字列を返します。<br>詳細については、「AppInfo (APP) 接続パラメーター」321 ページを参照してください。 |
| **UserDefinedCounterRate01** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターは、カウンター値の経時変化を表示します。<br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |

| プロパティ名 | 説明 |
| --- | --- |
| **UserDefinedCounterRate02** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターは、カウンター値の経時変化を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |
| **UserDefinedCounterRate03** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターは、カウンター値の経時変化を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |
| **UserDefinedCounterRate04** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターは、カウンター値の経時変化を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |
| **UserDefinedCounterRate05** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターは、カウンター値の経時変化を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **UserDefinedCounterRaw01** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターはカウンターの絶対値を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |
| **UserDefinedCounterRaw02** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターはカウンターの絶対値を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |
| **UserDefinedCounterRaw03** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターはカウンターの絶対値を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |
| **UserDefinedCounterRaw04** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターはカウンターの絶対値を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |

| プロパティ名 | 説明 |
| --- | --- |
| **UserDefinedCounterRaw05** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターはカウンターの絶対値を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |
| **UserID** | 接続のユーザー ID を返します。 |
| **UtilCmdsPermitted** | この接続で CREATE DATABASE、DROP DATABASE、RESTORE DATABASE などのユーティリティコマンドの使用が許可されているかどうかを示す値として、On または Off を返します。「-gu dbeng12/dbsrv12 サーバーオプション」247 ページを参照してください。 |
| **uuid_has_hyphens** | このプロパティでは、ユニークな識別子の値を文字列に変換するときのフォーマットを設定します。「uuid_has_hyphens オプション」707 ページを参照してください。 |
| **verify_password_function** | パスワードの検証に使用される関数が指定されている場合は、その関数名を返します。「verify_password_function オプション」708 ページを参照してください。 |
| **wait_for_commit** | データベースが次の COMMIT 文まで外部キー整合性をチェックしない場合は、On を返します。それ以外の場合は Off を返します。この場合、check_on_commit オプションで作成されていないすべての外部キーが、挿入、更新、削除時にチェックされます。<br>「wait_for_commit オプション」711 ページを参照してください。 |
| **WaitStartTime** | 接続が待機を始めた時刻 (接続が待機していない場合は空の文字列) を返します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **WaitType** | 待機の理由を返します (ある場合)。WaitType には次の値があります。<br>● **lock**　接続がロックを待っている場合に返されます。<br>● **waitfor**　接続が WAITFOR 文を実行している場合に返されます。<br>● **空の文字列**　接続が待機していないか、待機の理由が不明な場合に返されます。 |
| **webservice_namespace_host** | 生成された WSDL ドキュメント内で XML ネームスペースとして使用されるホスト名を返す (そのようなホスト名が指定されている場合)。「webservice_namespace_host オプション」712 ページを参照してください。 |

# データベースサーバープロパティ値のアクセス

データベースプロパティは、データベースサーバー全体に適用できます。PROPERTY システム関数を使用して個々のプロパティの値を取得するか、sa_eng_properties システムプロシージャーを使用してすべてのデータベースサーバープロパティの値を取得できます。プロパティ名では大文字と小文字が区別されません。

**♦ データベースサーバープロパティの値を取り出すには、次の手順に従います。**

● PROPERTY システム関数を使用します。たとえば、次の文はグローバルサーバーデータ構造に使用されたキャッシュページ数を返します。

```
SELECT PROPERTY ( 'MainHeapPages' );
```

**♦ すべてのデータベースサーバープロパティの値を取り出すには、次の手順に従います。**

● sa_eng_properties システムプロシージャーを使用します。

```
CALL sa_eng_properties;
```

**参照**


● 「PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』
● 「接続プロパティ値のアクセス」713 ページ
● 「データベースプロパティ値のアクセス」781 ページ

## データベースサーバープロパティ

| プロパティ名 | 説明 |
|---|---|
| **ActiveReq** | 現在クライアント側の要求を処理中のサーバーワーカーの数を返します。 |
| **ApproximateCPUTime** | データベースサーバーで使用された累積 CPU 時間の概算値 (秒単位) を返します。この値と実際の CPU 時間との誤差は、通常は 5 ～ 10 % ですが、最大で 50 % に達することもあります。マルチプロセッサーコンピューターでは、CPU (ハイパースレッドまたはコア) ごとに使用時間が積算されるため、すべての接続の累積時間を合計した値が実際の経過時間より大きくなる場合もあります。このプロパティは Windows と Linux でサポートされています。 |
| **AutoMultiProgrammingLevel** | データベースサーバーがマルチプログラミングレベルを自動的に調整しているかどうかを示す値を返します。「-gna dbsrv12 サーバーオプション」238 ページを参照してください。 |
| **AutoMultiProgrammingLevelStatistics** | データベースサーバーのマルチプログラミングレベルの自動調整に関するメッセージが、データベースサーバーのメッセージログに表示されるかどうかを示す値を返します。「-gns dbsrv12 サーバーオプション」241 ページを参照してください。 |
| **AvailIO** | 現在使用可能な I/O 制御ブロックの数を返します。 |
| **BuildChange** | このプロパティはシステムで使用するために予約されています。このプロパティの設定は変更しないでください。 |
| **BuildClient** | このプロパティはシステムで使用するために予約されています。このプロパティの設定は変更しないでください。 |
| **BuildProduction** | データベースサーバーが、実際の運用で使用するようにコンパイルされている場合は Yes を返し、デバッグ用のビルドである場合は No を返します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
|---|---|
| **BuildReproducible** | このプロパティはシステムで使用するために予約されています。このプロパティの設定は変更しないでください。 |
| **BytesReceived** | クライアント／サーバー通信中に受信したバイト数を返します。この値は、HTTP 接続と HTTPS 接続の場合に更新されます。 |
| **BytesReceivedUncomp** | 圧縮が無効になっている場合にクライアント／サーバー通信中に受信したバイト数を返します (この値は、圧縮が無効の場合は BytesReceived の値と同じ)。 |
| **BytesSent** | クライアント／サーバー通信中に送信されたバイト数を返します。この値は、HTTP 接続と HTTPS 接続の場合に更新されます。 |
| **BytesSentUncomp** | 圧縮が無効になっている場合にクライアント／サーバー通信中に送信されたバイト数を返します (この値は、圧縮が無効の場合は BytesSent の値と同じ)。 |
| **CacheAllocated** | サーバーデータ構造に割り付けられたキャッシュページの数を返します。 |
| **CacheFile** | データベースファイルから取得したデータを格納するキャッシュページの数を返します。 |
| **CacheFileDirty** | ダーティな (書き込みが必要な) キャッシュページの数を返します。 |
| **CacheFree** | 未使用のキャッシュページ数を返します。 |
| **CacheHits** | データベースページのルックアップ回数を返します。 |
| **CachePanics** | キャッシュマネージャーが割り付けるページの検索に失敗した回数を返します。 |
| **CachePinned** | 固定キャッシュページ数を返します。 |
| **CacheRead** | キャッシュの読み込み数を返します。 |
| **CacheReplacements** | キャッシュ内の置換されたページの数を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **CacheScavenges** | キャッシュマネージャーが割り付けるページをスカベンジした回数を返します。 |
| **CacheScavengeVisited** | 割り付けるページのスカベンジ中にアクセスしたページの数を返します。 |
| **CacheSizingStatistics** | キャッシュのサイズを変更するときにサーバーがキャッシュサイズ設定の統計を表示する場合は Yes を返し、それ以外の場合は No を返します。「-cs dbeng12/dbsrv12 サーバーオプション」215 ページを参照してください。 |
| **CarverHeapPages** | クエリ最適化など、短期的な目的に使用されたヒープページの数を返します。 |
| **CharSet** | データベースサーバーが使用している CHAR 文字セットを返します。 |
| **ClientStmtCacheHits** | クライアントに文がキャッシュされていることが理由で、不要な準備文の数を返します。これは、クライアントでの文のキャッシュが無効になった場合に必要となる、追加の準備文の数です。「max_client_statements_cached オプション」640 ページを参照してください。 |
| **ClientStmtCacheMisses** | クライアントのキャッシュに存在した文の中で、再び準備された文の数を返します。これは、キャッシュされた文が再使用できるか検討されたものの、スキーマの変更、データベースのオプション設定の変更、DROP VARIABLE 文によって再使用できなかった回数です。「max_client_statements_cached オプション」640 ページを参照してください。 |
| **CollectStatistics** | データベースサーバーがパフォーマンス統計を収集するかどうかを示す値として、Yes または No を返します。「-k dbeng12/dbsrv12 サーバーオプション」249 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **CommandLine** | データベースサーバーの起動に使用されたコマンドライン引数を返します。<br><br>-ek オプションを使用してデータベース用の暗号化キーを指定した場合、キーはこのプロパティで返される値にあるアスタリスクの定数文字列で置き換えられます。 |
| **Commit** | 処理されたコミット要求の数を返します。 |
| **CompactPlatformVer** | PlatformVer プロパティの縮小バージョンを返します。 |
| **CompanyName** | このソフトウェアを所有している会社の名前を返します。 |
| **ConnectedTime** | すべての接続が、これまでデータベースサーバーに接続してきた合計の時間数を秒単位で返します。<br><br>要求が接続に対して完了したとき、または接続が切断されたときにだけ値が更新されます。その結果、値は、データベースサーバー内で長いことアイドル状態であるか、実行中でビジーである接続に対しては遅れる可能性があります。値には、データベースイベントやバックグラウンドサーバー接続 (データベースクリーナーなど) といった任意の接続によって増えた時間も含まれます。 |
| **ConnsDisabled** | 新しい接続を禁止するサーバーオプションの現在の設定を示す値として、Yes または No を返します。「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **ConsoleLogFile** | -o オプションが指定されている場合は、データベースサーバーメッセージがロギングされるファイルの名前を返します。それ以外の場合は、空の文字列を返します。「-o dbeng12/dbsrv12 サーバーオプション」258 ページと「データベースサーバー動作のロギング」153 ページを参照してください。 |

| プロパティ名 | 説明 |
| --- | --- |
| **ConsoleLogMaxSize** | データベースサーバーメッセージのロギングに使用するファイルの最大サイズ (バイト単位) を返します。「-os dbeng12/dbsrv12 サーバーオプション」261 ページを参照してください。 |
| **CurrentCacheSize** | 現在のキャッシュサイズ (キロバイト単位) を返します。 |
| **CurrentMultiProgrammingLevel** | データベースサーバーが同時に処理できる現在のタスク数を返します。「マルチプログラミングレベルのデータベースサーバー設定」165 ページを参照してください。 |
| **Cursor** | 現在データベースサーバーが保持している宣言されたカーソルの数を返します。 |
| **CursorOpen** | 現在データベースサーバーが保持している開いたカーソルの数を返します。 |
| **DebuggingInformation** | サーバーがトラブルシューティング用の診断メッセージを表示する場合は Yes を返し、それ以外の場合は No を返します。「-z dbeng12/dbsrv12 サーバーオプション」296 ページを参照してください。 |
| **DefaultCollation** | 新規データベースに使用される照合を返します (明示的に指定されている照合がない場合)。 |
| **DefaultNcharCollation** | サーバーコンピューターのデフォルト NCHAR 照合の名前を返します (ICU がインストールされている場合は UCA、インストールされていない場合は UTF8BIN)。 |
| **DiskRead** | ディスクの読み込み数を返します。 |
| **DiskReadHintScatterLimit** | 分散読み込みヒントのサイズ制限 (バイト単位) を返します。 |
| **DiskRetryRead** | ディスクの読み込みリトライ数を返します。 |
| **DiskRetryReadScatter** | 分散読み込みのためのディスク読み込みリトライ数を返します。 |
| **DiskRetryWrite** | ディスクの書き込みリトライ数を返します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **EventTypeDesc** | 特定のイベントタイプ ID に関連付けられたシステムイベントタイプの説明を返します。 |
| **EventTypeName** | 特定のイベントタイプ ID に関連付けられたシステムイベントタイプの名前を返します。 |
| **ExchangeTasks** | クエリの並列実行に現在使用されているタスクの数を返します。 |
| **ExchangeTasksCompleted** | データベースサーバーが起動された後、クエリ内並列処理に使用された内部タスクの総数を返します。「高度：クエリ実行時の並列処理」『SQL Anywhere サーバー SQL の使用法』を参照してください。 |
| **FipsMode** | データベースサーバーの起動時に -fips オプションが使用された場合は Yes を返し、それ以外の場合は No を返します。 |
| **FirstOption** | データベースオプションに対応する最初の接続プロパティを表す番号を返します。 |
| **FreeBuffers** | 使用できるネットワークバッファーの数を返します。 |
| **FunctionMaxParms** | 関数で指定可能なパラメーターの最大数を返します。関数は、正の整数である *function-number* によって指定される値で識別されます。次に例を示します。<br>SELECT PROPERTY ( 'FunctionMaxParms', function-number );<br>*function-number* は、リリース間で変更されることがあります。 |
| **FunctionMinParms** | 関数で指定しなければならないパラメーターの最小数を返します。関数は、正の整数である *function-number* によって指定される値で識別されます。次に例を示します。<br>SELECT PROPERTY ( 'FunctionMinParms', function-number );<br>*function-number* は、リリース間で変更されることがあります。 |

| プロパティ名 | 説明 |
| --- | --- |
| **FunctionName** | 正の整数である *function-number* によって指定される値で識別される関数名を返します。<br><br>SELECT PROPERTY ( 'FunctionName', function-number );<br><br>*function-number* は、リリース間で変更されることがあります。 |
| **HeapsCarver** | クエリ最適化などの短期的な目的に使用されたヒープの数を返します。 |
| **HeapsLocked** | キャッシュ内で現在ロックされている再配置可能なヒープの数を返します。 |
| **HeapsQuery** | クエリ処理 (ハッシュ操作とソート操作) に使用されるヒープの数を返します。 |
| **HeapsRelocatable** | 再配置可能なヒープの数を返します。 |
| **HttpAddresses** | サーバーが HTTP 接続で受信するセミコロンで区切られた TCP/IP アドレスのリストを返します。次に例を示します。<br><br>(::1):80;127.0.0.1:80 |
| **HttpNumActiveReq** | HTTP 要求をアクティブに処理している HTTP 接続数を返します。応答を送信した HTTP 接続は含まれません。 |
| **HttpNumConnections** | データベースサーバーで現在開いている HTTP 接続数を返します。この接続は、要求をアクティブに処理しているか、長期 (keep-alive) 接続のキューで待機中である可能性があります。 |
| **HttpNumSessions** | データベースサーバー内のアクティブおよび休止状態の HTTP セッション数を返します。 |
| **HttpPorts** | Web サーバーの HTTP ポート番号をカンマで区切られたリストで返します。 |
| **HttpsAddresses** | サーバーが HTTPS 接続で受信するセミコロンで区切られた TCP/IP アドレスのリストを返します。次に例を示します。<br><br>(::1):443;127.0.0.1:443 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **HttpsNumActiveReq** | HTTPS 要求をアクティブに処理しているセキュア HTTPS 接続数を返します。応答を送信した HTTPS 接続は含まれません。 |
| **HttpsNumConnections** | データベースサーバーで現在開いている HTTPS 接続数を返します。この接続は、要求をアクティブに処理しているか、長期 (keep-alive) 接続のキューで待機中である可能性があります。 |
| **HttpsPorts** | Web サーバーの HTTPS ポート番号をカンマで区切られたリストで返します。 |
| **IdleTimeout** | デフォルトのアイドルタイムアウトを返します。「-ti dbeng12/dbsrv12 サーバーオプション」277 ページを参照してください。 |
| **IPAddressMonitorPeriod** | データベースサーバーが新しい IP アドレスをチェックする時間 (秒単位) を返します。「-xm dbeng12/dbsrv12 サーバーオプション」293 ページを参照してください。 |
| **IsEccAvailable** | ECC DLL がインストールされている場合は Yes を返し、それ以外の場合は No を返します。 |
| **IsFipsAvailable** | FIPS 認定の DLL がインストールされている場合は Yes を返し、それ以外の場合は No を返します。 |
| **IsIQ** | このプロパティはシステムで使用するために予約されています。このプロパティの設定は変更しないでください。 |
| **IsNetworkServer** | ネットワークデータベースサーバーに接続している場合は Yes を返し、パーソナルデータベースサーバーに接続している場合は No を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **IsPortableDevice** | データベースサーバーがラップトップ、ノートブック、またはその他のポータブルデバイスで実行されている場合は Yes、それ以外の場合は No を返します。Windows Mobile では常に Yes が返されます。VMWare は考慮されないため、ラップトップで実行されている VM で実行中のデータベースサーバーでは No が返されます。<br><br>Windows では、デバイスがポータブルかどうかを判別できない場合、このプロパティは NULL を返します。<br><br>UNIX の場合、このプロパティは常に NULL を返します。 |
| **IsRsaAvailable** | RSA DLL がインストールされている場合は Yes を返し、それ以外の場合は No を返します。 |
| **IsRuntimeServer** | データベースサーバーのすべてのバージョンについて No を返します。 |
| **IsService** | データベースサーバーがサービスとして実行されている場合は Yes、それ以外の場合は No を返します。 |
| **Language** | サーバーのロケール言語を返します。 |
| **LastConnectionProperty** | 最後の接続プロパティを表す番号を返します。 |
| **LastDatabaseProperty** | 最後のデータベースプロパティを表す番号を返します。 |
| **LastOption** | データベースオプションに対応する最後の接続プロパティを表す番号を返します。 |
| **LastServerProperty** | 最後のサーバープロパティを表す番号を返します。 |
| **LegalCopyright** | ソフトウェアの著作権を示す文字列を返します。 |
| **LegalTrademarks** | ソフトウェアの商標を示す文字列を返します。 |

| プロパティ名 | 説明 |
|---|---|
| **LicenseCount** | ライセンスされたシートまたはプロセッサーの数を返します。 |
| **LicensedCompany** | ライセンスされた会社の名前を返します。 |
| **LicensedUser** | ライセンスされたユーザーの名前を返します。 |
| **LicenseKey** | 内部でのみ使用。 |
| **LicenseType** | ライセンスタイプを返します。ネットワークシート (per-seat) または CPU ベースのいずれかです。 |
| **LivenessTimeout** | クライアント活性タイムアウトのデフォルトを返します。「-tl dbeng12/dbsrv12 サーバーオプション」278 ページを参照してください。 |
| **LockedCursorPages** | メモリ内でカーソルヒープを固定するために使用されているページ数を返します。 |
| **LockedHeapPages** | キャッシュ内でロックされているヒープページの数を返します。 |
| **MachineName** | データベースサーバーを実行しているコンピューターの名前を返します。通常は、コンピューターのホスト名です。 |
| **MainHeapBytes** | グローバルサーバーデータ構造に使用されているバイト数を返します。 |
| **MainHeapPages** | グローバルサーバーデータ構造に使用されているページ数を返します。 |
| **MapPhysicalMemoryEng** | Address Windowing Extensions を使用して、キャッシュ内の物理メモリにマッピングされているデータベースページのアドレス領域ウィンドウの数を返します。 |
| **MaxCacheSize** | 許容最大キャッシュサイズ (キロバイト単位) を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **MaxConnections** | サーバーが許容する同時接続の最大数を返します。デフォルト値は、パーソナルサーバーの場合は 10、ネットワークサーバーの場合は約 32000 です。この値を小さくするには、-gm サーバーオプションを使用します。「-gm dbeng12/dbsrv12 サーバーオプション」237 ページを参照してください。<br><br>通常は、コンピューターリソースの制約から、ネットワークサーバーへの接続の最大数はデフォルト値より小さくなります。 |
| **MaxEventType** | 有効な最大イベントタイプ ID を返します。 |
| **MaxMessage** | このプロパティは使用されなくなりました。データベースサーバーメッセージウィンドウから取得できる行番号の現在の最大値を返します。この値は、データベースサーバーメッセージウィンドウに最後に表示されたメッセージを表します。 |
| **MaxMultiProgrammingLevel** | データベースサーバーが同時に処理できるタスクの最大数を返します。「マルチプログラミングレベルのデータベースサーバー設定」165 ページを参照してください。 |
| **MaxRemoteCapability** | 有効な最大機能 ID を返します。 |
| **Message** | このプロパティは使用されなくなりました。データベースサーバーメッセージウィンドウから取得したメッセージ行を返します。行の先頭にはメッセージが表示された日付と時刻が追加されています。2 番目のパラメーターは行番号を指定します。<br><br>PROPERTY( "message" ) によって返された値が、データベースサーバーメッセージウィンドウに書き込まれた出力の最初の行です。PROPERTY( "message", n ) を呼び出すと、サーバー出力の n 行目 (0 を最初の行とする) が返されます。バッファーは有限であるため、メッセージの生成とともに最初の行が削除されていき、メモリからなくなる場合があります。この場合は、NULL が返されます。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **MessageCategoryLimit** | sa_server_messages システムプロシージャーを使用して取得できる重要度とカテゴリそれぞれの最小メッセージ数を返します。デフォルト値は 400 です。「sa_server_messages システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **MessageText** | このプロパティは使用されなくなりました。データベースサーバーメッセージウィンドウ内の指定された行番号に対応するテキスト (日付と時刻を含まない) を返します。2 番目のパラメーターは行番号を指定します。 |
| **MessageTime** | このプロパティは使用されなくなりました。データベースサーバーメッセージウィンドウ内の指定された行番号に対応する日付と時刻を返します。2 番目のパラメーターは行番号を指定します。 |
| **MessageWindowSize** | このプロパティは使用されなくなりました。データベースサーバーメッセージウィンドウから取得できる行の最大数を返します。 |
| **MinCacheSize** | 許容最小キャッシュサイズ (キロバイト単位) を返します。 |
| **MinMultiProgrammingLevel** | サーバーが同時に処理できるタスクの最小数を返します。「マルチプログラミングレベルのデータベースサーバー設定」165 ページを参照してください。 |
| **MultiPacketsReceived** | クライアント／サーバー通信中に受信したマルチパケット要求の数を返します。 |
| **MultiPacketsSent** | クライアント／サーバー通信中に送信されたマルチパケット応答の数を返します。 |
| **MultiPageAllocs** | マルチページキャッシュ割り付けの数を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **MultiProgrammingLevel** | サーバーが一度に処理できる同時タスクの最大数を返します。この値を超える数の同時タスクが存在する場合、要求はキューイングされます。この動作を変更するには、-gn サーバーオプションを使用します。「-gn dbsrv12 サーバーオプション」237 ページを参照してください。 |
| **Name** | データベースへの接続に使用するサーバーの別名を返す (別名が指定されている場合)。別名が指定されていない場合は、実際のサーバー名を返します。「-sn dbsrv12 オプション」315 ページと ServerName サーバープロパティ776 ページを参照してください。<br><br>クライアントがコピーノードに接続され、接続文字列で NodeType=COPY が指定されている場合、このプロパティの値は、ServerName 接続パラメーターでクライアントの接続文字列に指定されたデータベースサーバー名と異なることがあります。「NodeType (NODE) 接続パラメーター」357 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **NativeProcessorArchitecture** | プロセッサーをエミュレートできるプラットフォーム (X86 や Win64 など) のソフトウェアが実行されているネイティブプロセッサータイプを表す文字列を返します。これ以外の場合はすべて、ProcessorArchitecture プロパティと同じ値を返します。<br>ProcessorArchitecture サーバープロパティ 771 ページを参照してください。<br><br>X86、IA86、または X86_64 の場合は、値 X86 が返されることがあります。このプロパティでは、オペレーティングシステムが 32 ビットか 64 ビットかを示す値は返されません。<br><br>値には、次のものがあります。<br><br>● 32 ビット Windows (Windows Mobile 以外) − X86<br>● Windows Mobile − ARM<br>● 64 ビット Windows − X86_64<br>● Solaris − SPARC または X86_64<br>● AIX − PPC<br>● MAC OS − X86 または X86_64<br>● HP − IA64<br>● Linux − X86 または X86_64<br><br>サポートされるプラットフォームの詳細については、http://www.ianywhere.jp/sas/os.html を参照してください。 |
| **NumLogicalProcessors** | サーバーコンピューター上の論理プロセッサー数 (コアとハイパースレッドを含む) を返します。 |
| **NumLogicalProcessorsUsed** | データベースサーバーが使用する論理プロセッサーの数を返します。Windows の場合、使用する論理プロセッサー数を変更するには、-gtc オプションを使用します。「-gtc dbeng12/dbsrv12 サーバーオプション」 245 ページを参照してください。 |

| プロパティ名 | 説明 |
| --- | --- |
| **NumPhysicalProcessors** | サーバーコンピューターで有効となっている物理プロセッサーの数を返します。この値は、NumLogicalProcessors の値を物理プロセッサーあたりのコア数またはハイパースレッド数で割った数に相当します。Windows 以外のプラットフォームでは、コアまたはハイパースレッドが物理プロセッサーとして計数される場合があります。 |
| **NumPhysicalProcessorsUsed** | データベースサーバーが使用する物理プロセッサーの数を返します。パーソナルサーバーの場合、プラットフォームによってはプロセッサー数が 1 つに制限されることがあります。Windows でネットワークデータベースサーバーが使用する物理プロセッサーの数を変更するには、-gt オプションを使用します。「-gt dbeng12/dbsrv12 サーバーオプション」244 ページを参照してください。 |
| **ObjectType** | データベースオブジェクトのタイプを返します。この値は、SYSOBJECT システムビューによって使用されます。「SYSOBJECT システムビュー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **OmniIdentifier** | このプロパティはシステムで使用するために予約されています。このプロパティの設定は変更しないでください。 |
| **PacketsReceived** | 受信したクライアント／サーバー通信パケットの数を返します。この値は、HTTP 接続と HTTPS 接続の場合は更新されません。 |
| **PacketsReceivedUncomp** | 圧縮が無効になっている場合にクライアント／サーバー通信中に受信したパケットの数を返します(この値は、圧縮が無効の場合は PacketsReceived の値と同じ)。 |
| **PacketsSent** | 送信されたクライアント／サーバー通信パケットの数を返します。この値は、HTTP 接続と HTTPS 接続の場合は更新されません。 |
| **PacketsSentUncomp** | 圧縮が無効になっている場合にクライアント／サーバー通信中に送信されたパケットの数を返します(この値は、圧縮が無効の場合は PacketsSent の値と同じ)。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **PageSize** | データベースサーバーのキャッシュページのサイズを返します。-gp オプションを使用して設定可能です。設定しない場合はコマンドラインで指定したデータベースの最大ページサイズです。 |
| **PeakCacheSize** | 現在のセッションでキャッシュが到達した最大値 (キロバイト単位) を返します。 |
| **Platform** | ソフトウェアが実行されているオペレーティングシステムを返します。たとえば、Windows XP を実行中の場合、このプロパティは WindowsXP を返します。 |
| **PlatformVer** | ソフトウェアを実行しているオペレーティングシステムの名前を返します。この名前には、ビルド番号やサービスパックなどの情報も含まれています。 |
| **PrepStmt** | 現在データベースサーバーが保持している準備文の数を返します。 |
| **ProcessCPU** | データベースサーバープロセスにおける CPU の使用量を返します。値は秒数単位です。このプロパティは Windows と UNIX でサポートされています。このプロパティは Windows Mobile ではサポートされていません。<br><br>このプロパティが返す値は、データベースサーバーが起動してからの累積値です。この値は Windows タスク マネージャーや Windows パフォーマンス モニターなどのアプリケーションに表示される瞬時値とは一致しません。 |

| プロパティ名 | 説明 |
| --- | --- |
| **ProcessCPUSystem** | データベースサーバープロセス CPU におけるシステム CPU の使用量を返します。これは、データベースサーバーがオペレーティングシステムカーネルの内部で使用した CPU 時間を表します。値は秒数単位です。このプロパティは Windows と UNIX でサポートされています。このプロパティは Windows Mobile ではサポートされていません。<br><br>このプロパティが返す値は、データベースサーバーが起動してからの累積値です。この値は Windows タスク マネージャーやパフォーマンス モニターなどのアプリケーションに表示される瞬時値とは一致しません。 |
| **ProcessCPUUser** | データベースサーバープロセスにおけるユーザー CPU の使用量を返します。値は秒数単位です。これには、データベースサーバーがオペレーティングシステムカーネルの内部で使用した CPU 時間は含まれません。このプロパティは Windows と UNIX でサポートされています。このプロパティは Windows Mobile ではサポートされていません。<br><br>このプロパティが返す値は、データベースサーバーが起動してからの累積値です。この値は Windows タスク マネージャーやパフォーマンス モニターなどのアプリケーションに表示される瞬時値とは一致しません。 |
| **ProcessID** | データベースサーバープロセスのプロセス ID を返します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **ProcessorArchitecture** | 現在のソフトウェアが構築された対象のプロセッサータイプを表す文字列を返します。次のような値があります。<br><br>● 32 ビット Windows (Windows Mobile 以外) － X86<br>● 64 ビット Windows － X86_64<br>● Windows Mobile － ARM<br>● Solaris － SPARC または X86_64<br>● AIX － PPC<br>● MAC OS － X86<br>● HP － IA64<br>● Linux － X86 または X86_64<br><br>x86 および x64 のプラットフォームについて、このプロパティは 32 ビットのデータベースサーバーの場合 X86、64 ビットのデータベースサーバーの場合 X86_64 を返します。<br><br>詳細については、NativeProcessorArchitecture サーバープロパティ 767 ページを参照してください。 |
| **ProductName** | ソフトウェアの名前を返します。 |
| **ProductVersion** | 実行中のソフトウェアのバージョンを返します。 |
| **ProfileFilterConn** | 特定の接続をプロファイルするプロシージャーがオンの場合にモニターされる接続の ID を返します。それ以外の場合、空の文字列を返します。ユーザーによるプロシージャーのプロファイルは、sa_server_option プロシージャーを使って制御します。「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |

| プロパティ名 | 説明 |
| --- | --- |
| **ProfileFilterUser** | 特定のユーザーのプロシージャープロファイルがオンの場合にモニターされるユーザー ID を返します。それ以外の場合、空の文字列を返します。ユーザーによるプロシージャーのプロファイルは、sa_server_option プロシージャーを使って制御します。「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **QueryHeapPages** | クエリ処理 (ハッシュ操作とソート操作) に使用されるキャッシュページの数を返します。 |
| **QueryMemActiveCurr** | クエリメモリをアクティブに使用する要求数を返します。 |
| **QueryMemActiveEst** | 安定した状態のデータベースサーバーで、クエリメモリをアクティブに使用する要求の推定平均数を返します。 |
| **QueryMemActiveMax** | クエリメモリをアクティブに使用できる要求の最大数を返します。 |
| **QueryMemExtraAvail** | 基本的なメモリ集約型の付与を超えて付与可能なメモリ量を返します。 |
| **QueryMemGrantBase** | すべての要求に付与される最小メモリ量を返します。 |
| **QueryMemGrantBaseMI** | メモリ集約型の要求に付与される最小メモリ量を返します。 |
| **QueryMemGrantExtra** | アクティブなメモリ集約型の要求間で QueryMemGrantBaseMI の値を超えて分配されるクエリメモリページの数を返します。 |
| **QueryMemGrantFailed** | 要求がクエリメモリ待ちになり、取得できなかった合計回数を返します。 |
| **QueryMemGrantGranted** | 要求に現在付与されているページ数を返します。 |
| **QueryMemGrantRequested** | 要求がクエリメモリを取得しようとした合計回数を返します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
|---|---|
| **QueryMemGrantWaited** | 要求がクエリメモリ待ちになった合計回数を返します。 |
| **QueryMemGrantWaiting** | クエリメモリ待ちになっている現在の要求数を返します。 |
| **QueryMemPages** | クエリ実行アルゴリズムで使用可能なメモリの容量をページ数で返します。 |
| **QueryMemPercentOfCache** | クエリ実行アルゴリズムで使用可能なメモリの容量を最大キャッシュサイズの割合 (パーセント) で返します。 |
| **QuittingTime** | サーバーのシャットダウン時間を返します。シャットダウン時間が指定されていない場合は、none を返します。 |
| **RememberLastPlan** | オプティマイザーが返す最後のクエリ最適化プランをサーバーが記録する場合は、Yes を返します。「-zp dbeng12/dbsrv12 サーバーオプション」301 ページを参照してください。 |
| **RememberLastStatement** | サーバーが各接続で準備された最後の文を記録している場合は Yes を返し、それ以外の場合は No を返します。「-zl dbeng12/dbsrv12 サーバーオプション」297 ページを参照してください。 |
| **RemoteCapability** | 特定の機能 ID に関連付けられたリモート機能名を返します。 |
| **RemoteputWait** | 通信パケットの送信中にサーバーがブロックしなければならなかった回数を返します。通常ブロックは、クライアントまたはネットワークが受信可能な速度を超えてデータベースサーバーがデータを送信した場合にのみ起こります。エラー条件は示しません。 |
| **Req** | サーバーが新しい要求を処理するか、または既存の要求の処理を続行できる状態になった回数を返します。 |
| **ReqCountActive** | サーバーが新しい要求を処理するか、または既存の要求の処理を続行できる状態になった回数を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **ReqCountBlockContention** | 内部サーバーのリソースの競合により接続がブロックされた回数を返します。 |
| **ReqCountBlockIO** | IO 要求の完了を待つ間に接続がブロックされた回数を返します。 |
| **ReqCountBlockLock** | 別の接続が保持しているローロックを待つ間に接続がブロックされた回数を返します。 |
| **ReqCountUnscheduled** | サーバースレッドが接続を処理するのを待つ間に接続がブロックされた回数を返します。 |
| **ReqTimeActive** | サーバーが要求の処理で直接費やした合計時間を返します。 |
| **ReqTimeBlockContention** | 内部サーバーのリソースの競合により接続がブロックされた合計時間を返します。 |
| **ReqTimeBlockIO** | IO 要求の完了を待つ間に接続がブロックされた合計時間を返します。 |
| **ReqTimeBlockLock** | 別の接続が保持しているローロックを待つ間に接続がブロックされた合計時間を返します。 |
| **ReqTimeUnscheduled** | サーバースレッドが接続を処理するのを待つ間に接続がブロックされた合計時間を返します。 |
| **RequestFilterConn** | 接続のロギング情報をフィルターしている場合は、その接続の ID を返し、それ以外の場合は -1 を返します。 |
| **RequestFilterDB** | データベースのロギング情報をフィルターしている場合は、そのデータベースの ID を返し、それ以外の場合は -1 を返します。 |
| **RequestLogFile** | 要求ロギングファイルの名前を返します。要求がロギングされていない場合は、空の文字列を返します。「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **RequestLogging** | 現在の要求ロギング設定を示す値として、SQL、PLAN、HOSTVARS、PROCEDURES、TRIGGERS、OTHER、BLOCKS、REPLACE、ALL、NONE のいずれかを返します。「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **RequestLogMaxSize** | 要求ログファイルの最大サイズを返します。「-zs dbeng12/dbsrv12 サーバーオプション」303 ページを参照してください。 |
| **RequestLogNumFiles** | 保管される要求ログファイルの数を返します。「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **RequestsReceived** | クライアント／サーバー通信要求またはラウンドトリップの数を返します。このプロパティでは、PacketsReceived とは異なり、マルチパケット要求を 1 つの要求として数え、活性パケットを計数の対象から除外します。 |
| **RequestTiming** | 要求タイミングがオンになっている場合は Yes を返し、オフになっている場合は No を返します。要求タイミングをオンにするには、-zr データベースサーバーオプションを使用します。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |
| **Rlbk** | 処理されたロールバック要求の数を返します。 |
| **SendFail** | 通信プロトコルがパケット送信に失敗した回数を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **ServerEdition** | データベースサーバーのタイプを記述する単語のスペースで区切られたリストを返します。次のような値があります。<br><br>● Evaluation<br>● Developer<br>● Web<br>● Educational<br>● Standard<br>● Advanced<br>● Workgroup<br>● OEM<br>● Authenticated<br><br>次のいずれかの機能のためのライセンスを別途保有している場合は、返されるライセンス文字列に、次の該当する文字列が追加されます。<br><br>● **HighAvailability** 「SQL Anywhere の高可用性オプション」『SQL Anywhere 12 紹介』を参照してください。<br><br>● **InMemory** 「SQL Anywhere のインメモリモードオプション」『SQL Anywhere 12 紹介』を参照してください。<br><br>● **ECC** 「SQL Anywhere のセキュリティオプション」『SQL Anywhere 12 紹介』を参照してください。<br><br>● **FIPS** 「SQL Anywhere のセキュリティオプション」『SQL Anywhere 12 紹介』を参照してください。 |
| **ServerName** | 実際のサーバー名 (サーバーの代替名でない) を返します。この値によって、稼働しているサーバーのうちどれがデータベースミラーリング構成のプライマリサーバーであるかを特定できます。「データベースミラーリング」1137 ページと Name サーバープロパティ 766 ページを参照してください。 |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **StartDBPermission** | -gd サーバーオプションの設定を返します。値は、DBA、all、none のいずれかです。「-gd dbeng12/dbsrv12 サーバーオプション」233 ページを参照してください。 |
| **StartTime** | サーバーが起動された日付／時刻を返します。 |
| **StreamsUsed** | 使用中のデータベースサーバーストリーム数を返します。 |
| **TcpIpAddresses** | サーバーが Command Sequence と TDS 接続で受信するセミコロンで区切られた TCP/IP アドレスのリストを返します。次に例を示します。<br>(::1):2638;127.0.0.1:2638 |
| **TempDir** | テンポラリファイルが格納されているサーバー上のディレクトリを返します。 |
| **ThreadDeadlocksAvoided** | スレッドデッドロックエラーが検出されたがクライアントアプリケーションにレポートされなかった回数を返します。データベースサーバーが起動すると、このプロパティの値は 0 になります。<br><br>スレッドデッドロックエラーを返すのを防ぐため、データベースサーバーはマルチプログラミングレベルを動的に上げます。マルチプログラミングレベルを上げることができない場合、スレッドデッドロックエラーはクライアントアプリケーションに返され、ThreadDeadlocksReported プロパティが増分されます。パーソナルサーバー上ではマルチプログラミングレベルの動的な調整がデフォルトで無効になっているため、ThreadDeadlocksAvoided プロパティは常に 0 になります。<br><br>詳細については、「マルチプログラミングレベルのデータベースサーバー設定」165 ページを参照してください。 |

| プロパティ名 | 説明 |
| --- | --- |
| **ThreadDeadlocksReported** | スレッドデッドロックエラーがクライアントアプリケーションにレポートされた回数を返します。データベースサーバーが起動すると、このプロパティの値は 0 になります。 |
| **TimeZoneAdjustment** | サーバーのローカル時間を表示するために協定世界時 (UTC: Coordinated Universal Time) に加算する必要がある時間 (分単位) を返します。 |
| **TotalBuffers** | ネットワークバッファーの総数を返します。 |
| **UniqueClientAddresses** | ネットワークサーバーに接続しているユニークなクライアントネットワークアドレスの数を返します。 |
| **UnschReq** | 使用できるサーバーワーカーの解放を待ってキューイングされている要求の数を返します。 |
| **UserDefinedCounterRate01** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターは一定時間のカウンター値の変更を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |
| **UserDefinedCounterRate02** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターは、カウンター値の経時変化を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **UserDefinedCounterRate03** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターは、カウンター値の経時変化を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |
| **UserDefinedCounterRate04** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターは、カウンター値の経時変化を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |
| **UserDefinedCounterRate05** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターは、カウンター値の経時変化を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |
| **UserDefinedCounterRaw01** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターはカウンターの絶対値を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |

| プロパティ名 | 説明 |
| --- | --- |
| **UserDefinedCounterRaw02** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターはカウンターの絶対値を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |
| **UserDefinedCounterRaw03** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターはカウンターの絶対値を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |
| **UserDefinedCounterRaw04** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターはカウンターの絶対値を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |
| **UserDefinedCounterRaw05** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターはカウンターの絶対値を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **WebClientLogFile** | Web サービスクライアントログファイルの名前を返します。「-zoc dbeng12/dbsrv12 サーバーオプション」300 ページを参照してください。 |
| **WebClientLogging** | Web サービスクライアント情報がファイルにロギングされているかどうかを示す値を返します。「-zoc dbeng12/dbsrv12 サーバーオプション」300 ページを参照してください。 |

# データベースプロパティ値のアクセス

データベースプロパティは、データベースサーバーの各データベースで使用できます。DB_PROPERTY システム関数を使用して個々のプロパティの値を取得するか、sa_db_properties システムプロシージャーを使用してすべてのデータベースプロパティの値を取得できます。プロパティ名では大文字と小文字が区別されません。

**♦ データベースプロパティの値を取り出すには、次の手順に従います。**

- DB_PROPERTY システム関数を使用します。たとえば、次の文は、現在のデータベースのページサイズを返します。

```
SELECT DB_PROPERTY ( 'PageSize' );
```

**♦ すべてのデータベースプロパティの値を取り出すには、次の手順に従います。**

- sa_db_properties システムプロシージャーを使用します。

```
CALL sa_db_properties;
```

**参照**

- 「DB_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』
- 「DB_EXTENDED_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』
- 「データベースサーバープロパティ値のアクセス」753 ページ
- 「接続プロパティ値のアクセス」713 ページ

## データベースプロパティ

| プロパティ名 | 説明 |
|---|---|
| **AccentSensitive** | アクセント記号の区別のステータスを返します。データベースでアクセント記号が区別される場合は On、区別されない場合は Off、フランス語のアクセント記号の区別規則が適用されている場合は FRENCH を返します。 |
| **Alias** | データベース名を返します。 |
| **AlternateMirrorServerName** | データベースに関連付けられているミラーサーバーの代替名を返します (代替名が指定されている場合)。「-sm dbsrv12 データベースオプション (旧式)」314 ページを参照してください。 |
| **AlternateServerName** | データベースに関連付けられているサーバーの代替名を返します (代替名が指定されている場合)。「-sn dbsrv12 オプション」315 ページを参照してください。 |
| **ApproximateCPUTime** | データベースで使用された累積 CPU 時間の概算値 (秒単位) を返します。この値と実際の CPU 時間との誤差は、通常は 5 ～ 10 % ですが、最大で 50 % に達することもあります。マルチプロセッサーコンピューターでは、CPU (ハイパースレッドまたはコア) ごとに使用時間が積算されるため、すべての接続の累積時間を合計した値が実際の経過時間より大きくなる場合もあります。このプロパティは Windows と Linux でサポートされています。 |
| **ArbiterState** | 次のいずれかの値を返します。<br>● **NULL**　ミラーリングされていないデータベースに接続している。<br>● **connected**　監視サーバーはプライマリサーバーに接続している。<br>● **disconnected**　監視サーバーはプライマリサーバーに接続していない。<br>「データベースミラーリング」1137 ページを参照してください。 |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
|---|---|
| **AuditingTypes** | 現在有効な監査のタイプを返します。「auditing オプション」598 ページを参照してください。 |
| **Authenticated** | データベースが認証されている場合は Yes、データベースが認証されていない場合は No を返します。 |
| **BlankPadding** | データベースでブランク埋め込み機能が有効になっている場合は、On を返します。それ以外の場合は、Off を返します。 |
| **BytesReceived** | クライアント／サーバー通信中に受信したバイト数を返します。この値は、HTTP 接続と HTTPS 接続の場合に更新されます。 |
| **BytesReceivedUncomp** | 圧縮が無効になっている場合にクライアント／サーバー通信中に受信したバイト数を返します (この値は、圧縮が無効の場合は BytesReceived の値と同じ)。 |
| **BytesSent** | クライアント／サーバー通信中に送信されたバイト数を返します。この値は、HTTP 接続と HTTPS 接続の場合に更新されます。 |
| **BytesSentUncomp** | 圧縮が無効になっている場合にクライアント／サーバー通信中に送信されたバイト数を返します (この値は、圧縮が無効の場合は BytesSent の値と同じ)。 |
| **CacheHits** | キャッシュ内の検索において一致したデータベースページの数を返します。 |
| **CacheRead** | キャッシュ内で検索されたデータベースページの数を返します。 |
| **CacheReadIndInt** | キャッシュから読み込まれたインデックス内部ノードページの数を返します。 |
| **CacheReadIndLeaf** | キャッシュから読み込まれたインデックスリーフページの数を返します。 |
| **CacheReadTable** | キャッシュから読み込まれたテーブルページの数を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **CacheReadWorkTable** | キャッシュワークテーブルの読み込み数を返します。 |
| **Capabilities** | データベースに対して有効な機能ビットを返します。このプロパティは、主にテクニカルサポートを行うために使用されます。 |
| **CarverHeapPages** | クエリ最適化などの短期的な目的に使用されたヒープページの数を返します。 |
| **CaseSensitive** | 大文字と小文字の区別のステータスを返します。データベースで大文字と小文字が区別される場合は、On を返します。それ以外の場合は、Off を返します。大文字と小文字が区別されるデータベースでは、データの比較において大文字と小文字が異なるものとして扱われます。この設定は識別子における大文字と小文字の区別には影響しません。パスワードについては、常に大文字と小文字が区別されます。「大文字と小文字の区別」『SQL Anywhere サーバー SQL の使用法』を参照してください。 |
| **CatalogCollation** | カタログに使用される照合の識別子を返します。このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。「DB_EXTENDED_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **CharSet** | データベースの CHAR 文字セットを返します。<br><br>このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。「DB_EXTENDED_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **CheckpointLogCommitToDisk** | チェックポイントログのディスクへのコミット回数を返します。 |
| **CheckpointLogPagesInUse** | 使用中のチェックポイントログページ数を返します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **CheckpointLogPagesRelocated** | 再配置されたチェックポイントログページ数を返します。 |
| **CheckpointLogPagesWritten** | 書き込みが完了したチェックポイントログページ数を返します。 |
| **CheckpointLogSavePreimage** | チェックポイントログに追加されているデータベースページのプレイメージ数を返します。 |
| **CheckpointLogSize** | チェックポイントログのサイズ (ページ数) を返します。 |
| **CheckpointLogWrites** | チェックポイントログへの書き込み数を返します。 |
| **CheckpointUrgency** | 最後のチェックポイントからの経過時間を返します。この値は、データベースのチェックポイント時間の設定に対するパーセンテージで表されます。 |
| **Checksum** | データベースのグローバルチェックサムが有効になっている場合は、On を返します (チェックサムはデータベースページごとに作成されます)。それ以外の場合は、Off を返します。重要なページには必ずチェックサムが含まれています。 |
| **Chkpt** | 実行されたチェックポイントの数を返します。 |
| **ChkptFlush** | チェックポイント実行中に書き出された一連のページの数を返します。 |
| **ChkptPage** | トランザクションログチェックポイントの数を返します。 |
| **CleanablePagesAdded** | データベースサーバーの起動後、クリアされるようにマークが付けられたページの数を返します。 |
| **CleanablePagesCleaned** | データベースサーバーの起動後、クリアされたデータベースページの数を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **CleanableRowsAdded** | データベースサーバーの起動後、削除されるようにマークが付けられたローの数を返します。 |
| **CleanableRowsCleaned** | データベースサーバーの起動後、削除されたシャドウテーブルのローの数を返します。 |
| **ClientStmtCacheHits** | クライアントに文がキャッシュされていることが理由で、このデータベースには不要な準備文の数を返します。これは、クライアントでの文のキャッシュが無効になった場合に必要となる、追加の準備文の数です。<br>「max_client_statements_cached オプション」640 ページを参照してください。 |
| **ClientStmtCacheMisses** | このデータベースにおいて、クライアントのキャッシュに存在した文の中で、再び準備された文の数を返します。これは、キャッシュされた文が再使用できるか検討されたものの、スキーマの変更、データベースのオプション設定の変更、DROP VARIABLE 文によって再使用できなかった回数です。<br>「max_client_statements_cached オプション」640 ページを参照してください。 |
| **Collation** | データベースが使用する照合を返します。使用可能な照合のリストについては、「サポートされている照合と代替照合」492 ページを参照してください。<br><br>このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。「DB_EXTENDED_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **Commit** | ディスクキャッシュのフラッシュをサーバーが強制的に実行した回数を返します。<br>Windows では、バッファーなし (ダイレクト) IO を使用している場合、ディスクキャッシュのフラッシュは不要です。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **CommitFile** | ディスクキャッシュのフラッシュをサーバーが強制的に実行した回数を返します。Windows では、バッファーなし (ダイレクト) IO を使用している場合、ディスクキャッシュのフラッシュは不要です。 |
| **ConnCount** | データベースとの接続の数を返します。プロパティ値には、内部処理に使用されている接続は含まれませんが、イベントと外部環境サポートに使用されている接続は含まれます。<br><br>ライセンスされている接続の使用中の正確な数を取得するには、次の文を実行します。<br><br>SELECT COUNT( * ) FROM sa_conn_info( )<br>WHERE number < 100000000; |
| **ConnectedTime** | すべての接続が、これまでデータベースに接続してきた合計の時間数を秒単位で返します。<br><br>接続に対する要求が完了したとき、または接続が切断されたときにだけ値が更新されます。その結果、値は、データベースサーバー内で長いことアイドル状態であるか、実行中でビジーである接続に対しては遅れる可能性があります。値には、データベースイベントやバックグラウンドサーバー接続 (データベースクリーナーなど) といった任意の接続によって増えた時間も含まれます。 |
| **ConnPoolCachedCount** | アプリケーションによって切断されたが、接続プールにキャッシュされた接続の数を返します。 |
| **ConnPoolHits** | 再利用されたプール内の接続の数を返します。 |
| **ConnPoolMisses** | キャッシュされたが、再利用できなかった接続プール内の接続の数を返します。 |
| **ConnsDisabled** | 現在のデータベースとの接続が無効である場合は On を返し、それ以外の場合は Off を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **CurrentRedoPos** | 次のデータベース操作がロギングされるトランザクションログファイルの現在のオフセットを返します。 |
| **CurrIO** | サーバーが発行し、まだ完了していない現在のファイル I/O の数を返します。 |
| **CurrRead** | サーバーが発行し、まだ完了していない現在のファイル読み込み数を返します。 |
| **CurrWrite** | サーバーが発行し、まだ完了していない現在のファイル書き込み数を返します。 |
| **Cursor** | ディスクキャッシュのフラッシュをサーバーが強制的に実行した回数を返します。<br>Windows では、バッファーなし (ダイレクト) IO を使用している場合、ディスクキャッシュのフラッシュは不要です。 |
| **CursorOpen** | ディスクキャッシュのフラッシュをサーバーが強制的に実行した回数を返します。<br>Windows では、バッファーなし (ダイレクト) IO を使用している場合、ディスクキャッシュのフラッシュは不要です。 |
| **DatabaseCleaner** | データベースクリーナが有効になっているかどうかを示す On または Off を返します。 |
| **DBFileFragments** | データベースファイルのフラグメント数を返します。このプロパティは Windows でサポートされています。 |
| **DiskRead** | ディスクから読み込まれたページの数を返します。 |
| **DiskReadHint** | ディスクの読み込みヒント数を返します。 |
| **DiskReadHintPages** | ディスクの読み込みヒントページの数を返します。 |
| **DiskReadIndInt** | ディスクから読み込まれたインデックス内部ノードページの数を返します。 |
| **DiskReadIndLeaf** | ディスクから読み込まれたインデックスリーフページの数を返します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **DiskReadTable** | ディスクから読み込まれたテーブルページの数を返します。 |
| **DiskReadWorkTable** | ディスクワークテーブルの読み込み数を返します。 |
| **DiskRetryReadScatter** | 分散読み込みのためのディスク読み込みリトライ数を返します。 |
| **DiskSyncRead** | 同期的に発行されたディスクの読み込み数を返します。 |
| **DiskSyncWrite** | 同期的に発行された書き込み数を返します。 |
| **DiskWaitRead** | データベースサーバーによる非同期読み込みの待機回数を返します。 |
| **DiskWaitWrite** | データベースサーバーによる非同期書き込みの待機回数を返します。 |
| **DiskWrite** | ディスクに書き込まれた修正ページの数を返します。 |
| **DiskWriteHint** | ディスクの書き込みヒント数を返します。 |
| **DiskWriteHintPages** | ディスクの収集書き込みヒント数を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **DriveBus** | データベースファイルが接続されているバスのタイプを返します。値は次のいずれかです。<br><br>● UNKNOWN<br>● SCSI<br>● ATAPI<br>● ATA<br>● 1394<br>● SSA<br>● Fibre<br>● USB<br>● RAID<br>● iSCSI<br>● SAS<br>● SATA<br>● SD<br>● MMC<br>● Virtual<br>● FileBackedVirtual<br><br>バスのタイプを判別できない場合、このプロパティは NULL を返します。<br><br>プライマリ DB 領域に対するこのプロパティの値は SYSHISTORY ビューに記録されます。「SYSHISTORY システムビュー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **DriveModel** | データベースファイルが配置されているドライブのモデル番号を返します。プライマリ DB 領域に対するこのプロパティの値は SYSHISTORY ビューに記録されます。「SYSHISTORY システムビュー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。<br><br>ドライブのモデルを判別できない場合、このプロパティは NULL を返します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **DriveType** | データベースファイルが配置されているドライブのタイプを返します。値は次のいずれかです。<br><br>● CD<br>● FIXED<br>● RAMDISK<br>● REMOTE<br>● REMOVABLE<br>● UNKNOWN<br><br>UNIX では、UNIX のバージョンとドライブのタイプにより、ドライブタイプを判別できない場合があります。そのような場合、「UNKNOWN」が返されます。<br><br>このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。「DB_EXTENDED_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **Encryption** | データベース暗号化またはテーブル暗号化のタイプを返します。値は、None、Simple、AES、AES256、AES_FIPS、AES256_FIPS のいずれかです。 |
| **EncryptionScope** | データベース内に暗号化が可能な部分がある場合、その部分を示す値を返します。値は、TABLE、DATABASE、NONE のいずれかです。<br><br>TABLE は、テーブルの暗号化が有効になっていることを示します。DATABASE は、データベース全体が暗号化されることを示します。NONE は、テーブルの暗号化が有効になっておらず、データベースが暗号化されないことを示します。 |
| **ExprCacheAbandons** | ヒット率が低すぎるために式キャッシュが完全に中止された回数を返します。 |
| **ExprCacheDropsToReadOnly** | ヒット率が低いために式キャッシュが読み込み専用ステータスに変更された回数を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **ExprCacheEvicts** | 式キャッシュからの出力数を返します。 |
| **ExprCacheHits** | 式キャッシュ内のヒット数を返します。 |
| **ExprCacheInserts** | 式キャッシュに挿入された値の数を返します。 |
| **ExprCacheLookups** | 式キャッシュ内で実行されたルックアップの回数を返します。 |
| **ExprCacheResumesOfReadWrite** | ヒット率の上昇によって式キャッシュが読み込み／書き込みステータスに戻った回数を返します。 |
| **ExprCacheStarts** | 式キャッシュが開始された回数を返します。 |
| **ExtendDB** | データベースファイルを拡張するために追加されたページの数を返します。 |
| **ExtendTempWrite** | テンポラリファイルを拡張するために追加されたページの数を返します。 |
| **File** | パスを含むデータベースルートファイル名を返します。<br><br>このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。「DB_EXTENDED_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **FileSize** | SYSTEM DB 領域のファイルサイズ (ページ単位) を返します。このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。<br>「DB_EXTENDED_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **FreePages** | SYSTEM DB 領域の空きページ数を返します。FreePages プロパティは、バージョン 8.0.0 以降で作成されたデータベースでのみサポートされます。<br><br>このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。「DB_EXTENDED_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **FullCompare** | インデックスのハッシュ値を超えて実行された比較の回数を返します。 |
| **GetData** | GETDATA 要求の数を返します。 |
| **GlobalDBID** | レプリケーション環境でのユニークなプライマリキー値の生成に使用される global_database_id オプションの値を返します。 |
| **HasCollationTailoring** | データベースの作成時に照合の適合化が指定されていたかどうかを示す値を返します。使用できる値は On と Off です。 |
| **HasEndianSwapFix** | プラットフォームのエンディアンに関係なく、データベースがすべてのプラットフォームでビッグエンディアンとリトルエンディアンの両方の UTF-16 エンコードをサポートするかどうかを示す値を返します。使用できる値は On と Off です。 |
| **HashForcedPartitions** | メモリ競合が原因でハッシュ演算子が分割された回数を返します。 |
| **HashRowsFiltered** | ビットベクトルフィルターによって拒否されたプローブロー数を返します。 |
| **HashRowsPartitioned** | ハッシュワークテーブルに書き込まれたロー数を返します。 |
| **HashWorkTables** | ハッシュベースの演算用に作成されたワークテーブルの数を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **HasNCHARLegacyCollationFix** | 次のいずれかの値を返します。<br>● **ON**　バージョン 11 以降で作成されたデータベースと、従来の照合フィックスがインストールされ、従来の NCHAR 照合が使用されているバージョン 10 のデータベースサーバーで作成されたデータベースの場合。<br>● **OFF**　従来の照合フィックスがインストールされていないバージョン 10 のデータベースサーバーで作成されたデータベース、または従来の NCHAR 照合が使用されていないバージョン 10 のデータベースサーバーで作成されたデータベースの場合。 |
| **HasTornWriteFix** | 部分書き込みからのリカバリを可能にする固定ファイルフォーマットがデータベースにあるかどうかを示す値を返します。 |
| **HeapsCarver** | クエリ最適化などの短期的な目的に使用されたヒープの数を返します。 |
| **HeapsLocked** | キャッシュ内で現在ロックされている再配置可能なヒープの数を返します。 |
| **HeapsQuery** | クエリ処理 (ハッシュ操作とソート操作) に使用されるヒープの数を返します。 |
| **HeapsRelocatable** | 再配置可能なヒープの数を返します。 |
| **HttpConnPoolCachedCount** | すべてのプール内のキャッシュされたデータベース接続の絶対数を返します。 |
| **HttpConnPoolHits** | 同じサービスで再利用された接続の割合を返します。 |
| **HttpConnPoolMisses** | プールから接続にアクセスできなかったときに割り当てられた新しい接続の割合を返します。このプロパティでは、接続プールを使用するサービスへの HTTP 要求のみがカウントされます。アクセス時、プールのサイズが小さく、またそのプールの最も古い接続が横取りの基準に当てはまらなかったため、接続を使用できなかった可能性があります。 |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **HttpConnPoolSteals** | 別のサービスに接続が属するサービスによって利用された接続の割合を返します。<br><br>再利用可能な直接接続がないサービスの HTTP 要求および最も使用されていない接続のプールサイズと経過時間の条件を満たす場合、サービスは別のサービスから接続を横取りします。<br><br>プールの条件を満たさない場合は、サービスに接続が割り当てられ、HttpConnPoolMisses が増分されます。 |
| **IdentitySignature** | このプロパティは内部でのみ使用されます。 |
| **IdleCheck** | サーバーのアイドルスレッドがアクティブになり、アイドル書き込み、アイドルチェックポイントなどを実行した回数を返します。 |
| **IdleChkpt** | サーバーのアイドルスレッドが完了したチェックポイントの数を返します。アイドルスレッドが最後のダーティページをキャッシュに書き出すたびに、アイドルチェックポイントが発生します。 |
| **IdleChkTime** | アイドル I/O 中にチェックポイントに費やした 100 分の 1 秒単位の時間を返します。 |
| **IdleWrite** | サーバーのアイドルスレッドによるディスク書き込みの数を返します。 |
| **IndAdd** | インデックスに追加されたエントリの数を返します。 |
| **IndLookup** | インデックス内で検索されたエントリの数を返します。 |
| **IOParallelism** | DB 領域がサポートする同時 I/O 操作の推定回数を返します。このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。<br>「DB_EXTENDED_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |

| プロパティ名 | 説明 |
| --- | --- |
| **IOToRecover** | データベースのリカバリに必要な I/O 操作の推定回数を返します。 |
| **IQStore** | このプロパティは内部でのみ使用されます。 |
| **JavaVM** | データベースサーバーがデータベース内で Java を実行するために使用する Java VM を返します。 |
| **Language** | データベース照合によってサポートされている言語のカンマで区切ったリストを返します。言語は 2 文字の ISO フォーマットです。言語が不明である場合、このプロパティは NULL を返します。2 文字の ISO フォーマット言語名とそれに対応する言語のリストについては、「ロケール言語」485 ページを参照してください。 |
| **LastCheckpointTime** | 最新のチェックポイントの日時 (ミリ秒単位) を返します。 |
| **LastCommitRedoPos** | データベースによって最後の COMMIT 操作がトランザクションログに書き込まれた後で REDO ログの位置を返します。 |
| **LastSyncedRedoPos** | ディスクに対して書き込みが発行され、データが物理メディアに同期した最後の REDO の位置を返します。この位置よりも前にあるデータは、停電が発生した場合でもディスク上に存在すると考えられます。 |
| **LastWrittenRedoPos** | ディスクに対して書き込みが発行された最後の REDO の位置を返します。この書き込みはオペレーティングシステム、ディスクコントローラ、またはディスクドライブによってキャッシュできるため、物理メディアに必ずしも同期する必要はありません。 |
| **LockCount** | データベースで保持されているロックの数を返します。 |
| **LockTablePages** | ロック情報の保持に使用されているページの数を返します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **LogFileFragments** | ログファイルのフラグメント数を返します。このプロパティは Windows でサポートされています。 |
| **LogFreeCommit** | Redo Free Commit の数を返します。Redo Free Commit が発生するのは、トランザクションログのコミットが要求されているが、ログはすでに書き込まれている (コミットが実行されている) 場合です。 |
| **LogMirrorName** | パスを含むトランザクションログミラーのファイル名を返します。 |
| **LogName** | パスを含むトランザクションログファイル名を返します。 |
| **LogWrite** | トランザクションログに書き込まれたページの数を返します。 |
| **LTMGeneration** | このプロパティは内部でのみ使用されます。 |
| **LTMTrunc** | このプロパティは内部でのみ使用されます。 |
| **MaxIO** | CurrIO が到達した最大値を返します。 |
| **MaxRead** | CurrRead が到達した最大値を返します。 |
| **MaxWrite** | CurrWrite が到達した最大値を返します。 |
| **MirrorMode** | データベースミラーリングが使用中でない場合は NULL、-xp オプションで指定されたミラーリングモードが同期の場合は synchronous、それ以外の場合は asynchronous を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **MirrorRole** | 次のいずれかの値を返します。<br>● **primary**　プライマリデータベースサーバーまたはミラーサーバー。<br>● **mirror**　ミラーサーバー。<br>● **copy**　現在のプライマリサーバー、現在のミラーサーバー、または別の読み込み専用ノードからログページを取得する読み込み専用ミラーサーバー。コピーノードをプライマリサーバーにすることはできません。コピーノードは、ログページの取得元の親を持つため、子ノードとも呼ばれます。<br>「データベースミラーリング」1137 ページを参照してください。 |
| **MirrorServerState** | DB_EXTENDED_PROPERTY 関数で使用するときに、次の値の 1 つを返します。<br>● **NULL**　データベースはミラーリングされていません。<br>● **connected**　このサーバーから指定されたサーバーへの接続と、指定されたサーバーからこのサーバーへの接続があります。<br>● **disconnected**　このサーバーと指定されたサーバーとの間には接続はありません。<br>● **incoming only**　指定されたサーバーからこのサーバーへの接続があります。<br>● **outgoing only**　このサーバーから指定されたサーバーへの接続があります。<br>「データベースミラーリング」1137 ページと「DB_EXTENDED_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
|---|---|
| **MirrorServerWaits** | コピーサーバーへログページを送信するときにデータベースサーバーが 500 ミリ秒を超えて待機した回数を返します。 |
| **MirrorState** | 次のいずれかの値を返します。<br>● **NULL** ミラーリングされていないデータベースに接続している。<br>● **synchronizing** ミラーサーバーが接続されていないか、プライマリサーバーのログページの読み込みが完了していない。同期実行モードが非同期である場合にも、この値が返されます。<br>● **synchronized** ミラーサーバーが接続され、プライマリサーバー上でコミットされたすべての変更が反映されている。<br>「データベースミラーリング」1137 ページを参照してください。 |
| **MultiByteCharSet** | データベースで UTF-8 などのマルチバイト文字セットが使用されている場合は、On を返します。それ以外の場合は、Off を返します。詳細については、「マルチバイト文字セット」475 ページを参照してください。 |
| **Name** | データベース名を返します (Alias と同じ機能)。Alias データベースプロパティ 782 ページを参照してください。 |
| **NcharCharSet** | データベースの NCHAR 文字セットを返します。 |
| **NcharCollation** | NCHAR データに使用される照合の名前を返します。このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。「DB_EXTENDED_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |

| プロパティ名 | 説明 |
| --- | --- |
| **NextScheduleTime** | 指定したイベントの次の予定実行時間を返します。このプロパティに対するクエリには、DB_EXTENDED_PROPERTY 関数を使用します。「DB_EXTENDED_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **OptionWatchAction** | OptionWatchList プロパティに含まれているデータベースオプションを設定しようとしたときに実行されるアクションを返します。「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **OptionWatchList** | データベースサーバーによってモニターされているデータベースオプションのリストを返します。「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **PacketsReceived** | 受信したクライアント／サーバー通信パケットの数を返します。この値は、HTTP 接続と HTTPS 接続の場合は更新されません。 |
| **PacketsReceivedUncomp** | 圧縮が無効になっている場合にクライアント／サーバー通信中に受信したパケットの数を返します (この値は、圧縮が無効の場合は PacketsReceived の値と同じ)。 |
| **PacketsSent** | 送信されたクライアント／サーバー通信パケットの数を返します。この値は、HTTP 接続と HTTPS 接続の場合は更新されません。 |
| **PacketsSentUncomp** | 圧縮が無効になっている場合にクライアント／サーバー通信中に送信されたパケットの数を返します (この値は、圧縮が無効の場合は PacketsSent の値と同じ)。 |
| **PageRelocations** | テンポラリファイルから読み込まれた再配置可能なヒープページの数を返します。 |
| **PageSize** | データベースのページサイズ (バイト単位) を返します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **PartnerState** | 次のいずれかの値を返します。<br>● **NULL**　データベースはミラーリングされていません。<br>● **connected**　このサーバーからパートナーサーバーへの接続と、パートナーサーバーからこのサーバーへの接続があります。<br>● **disconnected**　このサーバーとパートナーサーバーとの間に接続はありません。<br>● **incoming only**　パートナーサーバーからこのサーバーへの接続があります。<br>● **outgoing only**　このサーバーからパートナーサーバーへの接続があります。<br>「データベースミラーリング」1137 ページを参照してください。 |
| **Prepares** | データベースに対して実行された文の準備作業の数を返します。 |
| **PrepStmt** | 現在データベースサーバーが保持している準備文の数を返します。 |
| **ProcedurePages** | プロシージャーで使用された再配置可能なヒープページの数を返します。 |
| **ProcedureProfiling** | データベースのプロシージャープロファイリングが有効になっている場合は、On を返します。それ以外の場合は、Off を返します。 |
| **QueryBypassed** | プランキャッシュから再利用された要求の数を返します。 |
| **QueryBypassedCosted** | オプティマイザーバイパスによってコストを使用して処理された要求の数を返します。 |
| **QueryBypassedHeuristic** | オプティマイザーバイパスによってヒューリスティックを使用して処理された要求の数を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **QueryBypassedOptimized** | オプティマイザーバイパスによって最初に処理されてから、SQL Anywhere オプティマイザーによって完全に最適化された要求の数を返します。 |
| **QueryCachedPlans** | すべての接続においてキャッシュされている実行プランの数を返します。 |
| **QueryCachePages** | 実行プランの保存に使用されているキャッシュページの数を返します。 |
| **QueryDescribedBypass** | オプティマイザーバイパスによって処理された DESCRIBE 要求の数を返します。 |
| **QueryDescribedOptimizer** | オプティマイザーによって処理された DESCRIBE 要求の数を返します。 |
| **QueryHeapPages** | クエリ処理 (ハッシュ操作とソート操作) に使用されるキャッシュページの数を返します。 |
| **QueryJHToJNLOptUsed** | ハッシュジョインがネストループジョインに変換された回数を返します。 |
| **QueryLowMemoryStrategy** | メモリ不足が原因でサーバーが実行プランを実行中に変更した回数を返します。使用できるメモリがオプティマイザーの推定よりも少ない、または実行プランが必要とするメモリがオプティマイザーの推定よりも多い場合に、プランが変更されることがあります。 |
| **QueryMemActiveCurr** | クエリメモリをアクティブに使用する要求数を返します。 |
| **QueryMemGrantFailed** | 要求がクエリメモリ待ちになり、取得できなかった合計回数を返します。 |
| **QueryMemGrantGranted** | 要求に現在付与されているページ数を返します。 |
| **QueryMemGrantRequested** | 要求がクエリメモリを取得しようとした合計回数を返します。 |
| **QueryMemGrantWaited** | 要求がクエリメモリ待ちになった合計回数を返します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
|---|---|
| **QueryMemGrantWaiting** | クエリメモリ待ちになっている現在の要求数を返します。 |
| **QueryOpened** | 実行対象の OPEN 要求の数を返します。 |
| **QueryOptimized** | 完全に最適化された要求の数を返します。 |
| **QueryReused** | 再利用されたクエリプラン数を返します。 |
| **QueryRowsFetched** | このデータベースについて、逐次スキャンまたはインデックススキャンによってベーステーブルから読み込まれたローの数を返します。 |
| **QueryRowsMaterialized** | クエリ処理中にワークテーブルに書き込まれるロー数を返します。 |
| **ReadOnly** | データベースが読み込み専用モードで実行されている場合は、On を返します。それ以外の場合は、Off を返します。 |
| **ReceivingTracingFrom** | トレーシングデータが保存されていたデータベースの名前を返します。トレーシングが追加されていない場合は、ブランク文字列を返します。 |
| **RecoveryUrgency** | データベースのリカバリに要する推定時間を返します。この値は、データベースのリカバリ時間設定に対する割合で表されます。「-gr dbeng12/dbsrv12 サーバーオプション」242 ページと「データベースサーバーがチェックポイントのタイミングを決定する方法」1114 ページを参照してください。 |
| **RecursiveIterations** | 再帰ユニオンの反復回数を返します。 |
| **RecursiveIterationsHash** | 再帰ハッシュジョインでハッシュ方式が使用された回数を返します。 |
| **RecursiveIterationsNested** | 再帰ハッシュジョインでネストループ方式が使用された回数を返します。 |
| **RecursiveJNLMisses** | 再帰ハッシュジョインでのインデックスプローブキャッシュミスの数を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **RecursiveJNLProbes** | 再帰ハッシュジョインにおけるインデックスプローブの試行回数を返します。 |
| **RelocatableHeapPages** | 再配置可能なヒープ (カーソル、文、プロシージャー、トリガー、ビューなど) で使用されるページの数を返します。 |
| **RemoteTrunc** | SQL Remote Message Agent 用に最後に確認されたログオフセットを返します。 |
| **ReqCountActive** | 処理が完了した要求の数を返します。RequestTiming サーバープロパティが Off に設定されている場合は、NULL を返します。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |
| **ReqCountBlockContention** | 接続がアトミックアクセスを待ったデータベースへの接続回数を返します。-zt オプションが指定されていない場合は、NULL を返します。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |
| **ReqCountBlockIO** | データベースへの接続が I/O 処理の完了を待った回数を返します。-zt オプションが指定されていない場合は、NULL を返します。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |
| **ReqCountBlockLock** | 接続がロックを待ったデータベースへの接続回数を返します。-zt オプションが指定されていない場合は、NULL を返します。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |
| **ReqCountUnscheduled** | 接続がスケジューリングを待ったデータベースへの接続回数を返します。-zt オプションが指定されていない場合は、NULL を返します。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |
| **ReqTimeActive** | 要求の処理に要した時間を返します。-zt オプションが指定されていない場合は、NULL を返します。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
|---|---|
| **ReqTimeBlockContention** | アトミックアクセスを取得するまでの待ち時間を返します。RequestTiming サーバープロパティが Off に設定されている場合は、NULL を返します。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |
| **ReqTimeBlockIO** | I/O 処理が完了するまでの待ち時間を返します。-zt オプションが指定されていない場合は、NULL を返します。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |
| **ReqTimeBlockLock** | ロックが解放されるまでの待ち時間を返します。-zt オプションが指定されていない場合は、NULL を返します。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |
| **ReqTimeUnscheduled** | 未スケジュール時間を返します。-zt オプションが指定されていない場合は、NULL を返します。「-zt dbeng12/dbsrv12 オプション」304 ページを参照してください。 |
| **RequestsReceived** | クライアント／サーバー通信要求またはラウンドトリップの数を返します。このプロパティでは、PacketsReceived とは異なり、マルチパケット要求を 1 つの要求として数え、活性パケットを計数の対象から除外します。 |
| **Rlbk** | 処理されたロールバック要求の数を返します。 |
| **RollbackLogPages** | ロールバックログのページ数を返します。 |
| **SendingTracingTo** | トレーシングデータの送信先を示す接続文字列を返します。トレーシングは追加されていない場合は、ブランク文字列を返します。 |
| **SnapshotCount** | データベースに関連付けられているスナップショットの数を返します。 |

| プロパティ名 | 説明 |
| --- | --- |
| **SnapshotIsolationState** | 次のいずれかの値を返します。<br>● **On**　データベースでスナップショットアイソレーションが有効になっている。<br>● **Off**　データベースでスナップショットアイソレーションが無効になっている。<br>● **in_transition_to_on**　現在のトランザクションの完了後にスナップショットアイソレーションが有効となる。<br>● **in_transition_to_off**　現在のトランザクションの完了後にスナップショットアイソレーションが無効となる。<br>「allow_snapshot_isolation オプション」590 ページを参照してください。 |
| **SortMergePasses** | ソート中に使用されたマージパスの数を返します。 |
| **SortRowsMaterialized** | ソートワークテーブルに書き込まれたロー数を返します。 |
| **SortRunsWritten** | ソート中に書き込まれたソート実行の数を返します。 |
| **SortSortedRuns** | 実行の生成中に作成されたソート実行の数を返します。 |
| **SortWorkTables** | ソート用に作成されたワークテーブルの数を返します。 |
| **StatementDescribes** | DESCRIBE 要求によって処理された文の合計数を返します。 |
| **StatementPostAnnotates** | セマンティックなクエリ変換フェーズによって処理された文の数を返します。 |
| **StatementPostAnnotatesSimple** | セマンティックなクエリ変換フェーズによって処理されたが、一部のセマンティック変換が省略された文の数を返します。 |
| **StatementPostAnnotatesSkipped** | セマンティックなクエリ変換フェーズが完全に省略された文の数を返します。 |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **SynchronizationSchemaChangeActive** | アクティブな接続によって START SYNCHRONIZATION SCHEMA CHANGE が発行されたが、STOP SYNCHRONIZATION SCHEMA CHANGE は発行されていない場合、On を返します。「START SYNCHRONIZATION SCHEMA CHANGE 文 [Mobile Link]」『SQL Anywhere サーバー SQL リファレンス』と「STOP SYNCHRONIZATION SCHEMA CHANGE 文 [Mobile Link]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **SyncTrunc** | Mobile Link クライアントの dbmlsync 実行プログラム用に最後に確認されたログオフセットを返します。 |
| **TempFileName** | パスを含むデータベーステンポラリファイル名を返します。 |
| **TempTablePages** | テンポラリテーブルで使用されるテンポラリファイルのページ数を返します。 |
| **TriggerPages** | トリガーで使用される再配置可能なヒープページの数を返します。 |
| **UserDefinedCounterRate01** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターは、カウンター値の経時変化を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |
| **UserDefinedCounterRate02** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターは、カウンター値の経時変化を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |

| プロパティ名 | 説明 |
| --- | --- |
| **UserDefinedCounterRate03** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターは、カウンター値の経時変化を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |
| **UserDefinedCounterRate04** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターは、カウンター値の経時変化を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |
| **UserDefinedCounterRate05** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターは、カウンター値の経時変化を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |
| **UserDefinedCounterRaw01** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターはカウンターの絶対値を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| プロパティ名 | 説明 |
| --- | --- |
| **UserDefinedCounterRaw02** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターはカウンターの絶対値を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |
| **UserDefinedCounterRaw03** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターはカウンターの絶対値を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |
| **UserDefinedCounterRaw04** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターはカウンターの絶対値を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |
| **UserDefinedCounterRaw05** | ユーザーが定義したパフォーマンスカウンターの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンターは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターはカウンターの絶対値を表示します。<br><br>詳細については、「ユーザー定義プロパティ」810 ページを参照してください。 |

| プロパティ名 | 説明 |
| --- | --- |
| **VersionStorePages** | スナップショットアイソレーションが有効な場合にローバージョンストアで使用されるテンポラリファイルのページ数を返します。 |
| **ViewPages** | ビューで使用される再配置可能なヒープページの数を返します。 |
| **WriteChecksum** | チェックサムが書き出される前にデータベースサーバーによってページに追加される場合は On を返し、それ以外の場合は Off を返します。「チェックサムを使用した破損の検出」1120 ページを参照してください。 |
| **XPathCompiles** | データベースサーバーの起動後、(openxml プロシージャー を使用する) XPath クエリがコンパイルされた回数を返します。「openxml システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |

# ユーザー定義プロパティ

SQL Anywhere は、ユーザー定義値に設定できるプロパティをサポートしています。アプリケーションでは、それぞれのユーザー定義プロパティのセマンティックを定義し、返される値を設定する必要があります。プロパティ値は、sa_user_defined_counter_set システムプロシージャーを使用して絶対値を設定したり、sa_user_defined_counter_add システムプロシージャーを使って増分することができます。sa_user_defined_counter_add システムプロシージャーは、負の値を加算して、プロパティを減分するためにも使用できます。

ユーザー定義プロパティの値は、それぞれの接続、データベース、サーバーで個別に管理されます。プロパティの現在の値は、CONNECTION_PROPERTY、DB_PROPERTY、および PROPERTY 関数をそれぞれに使うことで検索できます。これらのプロパティは、パフォーマンスモニターからもアクセスできます。パフォーマンスモニターは、UserDefinedCounterRaw プロパティの現在の値を絶対値で表示し、ある測定間隔内の UserDefinedCounterRate 変更速度を速度で表示します。

これらのプロパティを使用する例として、キープールに残っている現在のキーの数を示す未加工プロパティを持つことがあります。このプロパティの値は、時間の経過に従って上下に変動します。同時に速度プロパティもキープールからキーがどの程度の速さで使用されたかを示すことができます。この値は時間が経ってキーが使用されるにつれ増加し、パフォーマンスモニターはキーが使用された速度を秒単位で示します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**参照**

- UserDefinedCounterRaw01 接続プロパティ 751 ページ
- UserDefinedCounterRate01 接続プロパティ 749 ページ
- UserDefinedCounterRaw01 データベースプロパティ 808 ページ
- UserDefinedCounterRate01 データベースプロパティ 807 ページ
- UserDefinedCounterRaw01 サーバープロパティ 779 ページ
- UserDefinedCounterRate01 サーバープロパティ 778 ページ
- 「sa_user_defined_counter_add システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「sa_user_defined_counter_set システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「ユーザー定義統計情報」『SQL Anywhere サーバー SQL の使用法』
- 「統計のモニターに使用される SQL 関数」『SQL Anywhere サーバー SQL の使用法』
- 「Windows パフォーマンスモニターを使用した統計値のモニタリング」『SQL Anywhere サーバー SQL の使用法』

# SQL Anywhere のサイズと数の制限

SQL Anywhere データベースにおけるオブジェクトのサイズと数の制限について次の表に示します。一般に、コンピューターのメモリ、CPU、ディスク容量から受ける制限の方が厳しいのが普通です。

| 項目 | 制限 |
|---|---|
| データベースサイズ | 13 ファイル／データベース。オペレーティングシステムとファイルシステムが許可する各ファイルの最大サイズ。 |
| DB 領域サイズ | $2^{28}$ x ページサイズ |
| テンポラリファイルサイズ | $2^{28}$ x ページサイズ |
| フィールドの大きさ | 2 GB |
| ファイルサイズ (FAT 12) | 16 MB |
| ファイルサイズ (FAT 16) | 2 GB |
| ファイルサイズ (FAT 32) | 4 GB |
| ファイルサイズ (NTFS、HP-UX 11.0 以降、Solaris 2.6 以降、Linux 2.4 以降) | ●512 GB (2 KB ページに対して)<br>●1 TB (4 KB ページに対して)<br>●2 TB (8 KB ページに対して) |

| 項目 | 制限 |
|---|---|
| ファイルサイズ (その他のプラットフォームとファイルシステム) | 2 GB |
| 最大キャッシュサイズ (非 AWE キャッシュ) (Windows XP Home Edition、Windows XP Professional、Windows Server 2003 Web Edition、Windows Server 2003 Standard Edition、Windows Server 2008、Windows Server 2008 R2、Windows 7) | 1.8 GB |
| 最大キャッシュサイズ (非 AWE キャッシュ) (Windows Server 2003 Enterprise Edition、Windows Server 2003 Datacenter Edition、Windows Vista Ultimate、Windows Vista Enterprise、Windows Vista Business、Windows Vista Home Premium、Windows Vista Home Basic) | 2.7 GB |
| 最大キャッシュサイズ (AWE キャッシュ) (Windows XP Home Edition、Windows XP Professional、Windows Server 2003 Web Edition、Windows Server 2003 Standard Edition、Windows Server 2003 Enterprise Edition、Windows Server 2003 Datacenter Edition) | 使用可能な全メモリの 100% ～ 128 MB |
| 最大キャッシュサイズ (Windows Mobile) | デバイス上の使用可能メモリによる |
| 最大キャッシュサイズ (UNIX - Solaris、x86 Linux、IBM AIX、HP) | 2 GB (32 ビットサーバーに対して) |
| 最大キャッシュサイズ (Win 64) | 64 ビットサーバーの物理メモリによる |
| 最大キャッシュサイズ (Itanium HP-UX) | 64 ビットサーバーの物理メモリによる |
| 最大インデックスエントリサイズ | 制限なし |
| データベース数／サーバー | 255 |
| カラム数／テーブル | 45000<br>注意:カラム数を過度に多くすることは可能だがパフォーマンスに影響する。 |
| NULL 入力可のカラムの数／テーブル | 45000 と ((ページサイズ - オーバーヘッド) * 8) のうち小さい方 |
| プロシージャーの結果セット内のカラム数 | 45000 |
| SELECT リスト内のカラム数 | 100000 |

| 項目 | 制限 |
|---|---|
| GROUP BY リスト内のカラム数 | 100000 |
| グループセットのある GROUP BY リスト内のカラム数 | 64 |
| CUBE 内のカラム数 | 15 |
| 異なるグループセットの数 | 32768 |
| カラムの DEFAULT の長さ | 32768 |
| カラムの COMPUTE の長さ | 32768 |
| プロシージャーパラメーターの DEFAULT の長さ | 32768 |
| ユーザー定義ドメインの DEFAULT の長さ | 32768 |
| 検査制約の長さ | 2 GB |
| インデックス数／テーブル | $2^{32}$ |
| ロー数／データベース | 4096 x $2^{28}$ x 13 |
| ロー数／テーブル | 4096 x $2^{28}$ |
| テーブル数／データベース | $2^{32}$ - $2^{20}$ - 1 = 4293918719 |
| テンポラリテーブル数／接続 | $2^{20}$ = 1048576 |
| 参照されるテーブル数／トランザクション | 制限なし |
| ストアドプロシージャー数／データベース | $2^{32}$ - 1 = 4294967295 |
| 同時実行文の数／データベースサーバー | 20 x *データベース接続数* + 65534 |
| イベント数／データベース | $2^{31}$ - 1 = 2147483647 |
| トリガー数／データベース | $2^{32}$ - 1 = 4294967295 |
| ローサイズ | ファイルサイズにより制限 |
| テーブルの大きさ | 最大ファイルサイズ。テーブルのユーザー定義インデックスは、テーブルとは別に保存可能。 |
| 文字データ型 | $2^{31}$ - 1 = 2147483647 |

| 項目 | 制限 |
| --- | --- |
| バイナリデータ型 | $2^{31}$ - 1 = 2147483647 |
| 識別子 (ユーザー ID、テーブル名、カラム名を含む) | 128 バイト |
| パスワード | 255 バイト |
| データベースサーバー名 | 250 バイト (TCP/IP と共有メモリ)<br><br>「-n dbeng12/dbsrv12 サーバーオプション」257 ページと「ServerName (Server) 接続パラメーター」365 ページを参照してください。 |
| データベース名 | 250 バイト<br><br>「-n dbeng12/dbsrv12 データベースオプション」312 ページを参照してください。 |

# SQL Anywhere のハードウェア要件

ハードウェア要件の詳細については、次の URL を参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

# データベース管理

この項では、SQL Anywhere に付属のツールを使用してデータベースを管理する方法について説明します。

## SQL Anywhere グラフィカル管理ツール

## Sybase Central

Sybase Central は、データベースサーバー、データベース、およびそれらに含まれているオブジェクトを管理するためのグラフィカルツールです。

Sybase Central で **[ヘルプ]** » **[Sybase Central]** をクリックすると、Sybase Central の使用や設定についての詳細情報を参照できます。

### Sybase Central の主な機能

- **簡単なコマンドアクセス**　オブジェクトを選択すると Sybase Central の **[ファイル]** メニューが自動的に更新され、そのオブジェクトに直接関連するコマンドが表示されます。オブジェクトを右クリックしてこれらのコマンドにアクセスすることもできます。

- **タスクウィザード**　Sybase Central には、新しいオブジェクトを追加するための、段階を踏んでタスクを実行するウィザードが準備されています。

- **ドラッグアンドドロップ機能**　Sybase Central では、多くのオペレーションでドラッグアンドドロップ機能を利用できます。たとえば、異なるデータベースへテーブルをコピーするには、テーブルをクリックしてコピー先にドラッグします。「SQL Anywhere 12 プラグインのデータベースオブジェクト」831 ページを参照してください。

- **キーボードショートカット**　一般的に使用される多くのコマンドにはキーボードショートカットがあります。ショートカットは、メニューのコマンド名の横に表示されています。「Sybase Central キーボードショートカット」823 ページを参照してください。

- **プラグインサポート**　プラグインを使用して、さまざまなデータベース製品やツールを管理できます。Sybase Central で **[ヘルプ]** をクリックしてからプラグイン名を選択すると、そのプラグインの使用や設定についての詳細情報を参照できます。

### プラグイン

製品は、それぞれ異なるプラグインで管理されています。Sybase Central で製品を使用する前に、該当するプラグインを登録し、ロードする必要があります。プラグインは、製品のインストール時に自動的に登録され、ロードされます。

SQL Anywhere には、次の製品に対応する Sybase Central プラグインが備えられています。

- SQL Anywhere データベース
- Ultra Light データベース
- Mobile Link 同期
- QAnywhere メッセージ
- Relay Server

プラグインファイルは、SQL Anywhere インストール環境の以下のロケーションに格納されます。

| プラグイン | ファイル名とロケーション |
|---|---|
| **SQL Anywhere 12** | *%SQLANY12%¥java¥saplugin.jar* |
| **Mobile Link 12** | *%SQLANY12%¥java¥mlplugin.jar* |
| **Ultra Light 12** | *%SQLANY12%¥java¥ulplugin.jar* |
| **QAnywhere 12** | *%SQLANY12%¥java¥qaplugin.jar* |
| **Relay Server 12** | *%SQLANY12%¥java¥rsplugin.jar* |

### Sybase Central の配備

ライセンス契約に従って、Sybase Central を含む SQL Anywhere の管理ツールを配備できます。

アプリケーションとともに Sybase Central を配備する方法については、「管理ツールの配備」『SQL Anywhere サーバー プログラミング』を参照してください。

### 参照

- SQL Anywhere： 「SQL Anywhere 12 プラグイン」830 ページ
- Mobile Link： 「Sybase Central の Mobile Link プラグイン」『Mobile Link クイックスタート』
- Ultra Light：「データベース作成ウィザードでの Ultra Light データベースの作成」『Ultra Light データベース管理とリファレンス』
- QAnywhere： 「QAnywhere 12 プラグイン」『QAnywhere』
- Relay Server： 「Sybase Central 用の Relay Server プラグイン」『Relay Server』

## Sybase Central の起動 (Windows の場合)

Sybase Central は、データベースサーバ、データベース、含まれるオブジェクトを管理するため、SQL 文とコマンドラインユーティリティの代わりとして使用できるグラフィックツールです。

### 前提条件

この作業を実行するための前提条件は、ありません。

### 内容と備考

Windows (Windows Mobile を除く)

♦ **Sybase Central の起動とサンプルデータベースへの接続 (Windows の場合)**

1. **[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[Sybase Central]** をクリックします。
2. **[接続]** » **[SQL Anywhere 12 に接続]** をクリックします。
3. **[アクション]** ドロップダウンリストで、**[ODBC データソースを使用した接続]** を選択します。
4. **[ODBC データソース名]** をクリックし、その下のボックスに **SQL Anywhere 12 Demo** と入力します。
5. **[接続]** をクリックします。

**結果**

Sybase Central は現在、サンプルデータベースに接続されています。

**次の手順**

サンプルデータベースが起動したので、特定のオブジェクトについて、データベースを検索できます。

**参照**


- 「[接続] ウィンドウを開く (Sybase Central の場合)」14 ページ
- 「Sybase Central の起動とサンプルデータベースへの接続 (Mac OS X の場合)」817 ページ
- 「Sybase Central の起動とサンプルデータベースへの接続 (UNIX コマンドラインの場合)」818 ページ
- 「Sybase Central の起動とサンプルデータベースへの接続 (Linux の [アプリケーション] メニューの場合)」819 ページ


## Sybase Central の起動とサンプルデータベースへの接続 (Mac OS X の場合)

Sybase Central は、データベースサーバ、データベース、含まれるオブジェクトを管理するため、SQL 文とコマンドラインユーティリティの代わりとして使用できるグラフィックツールです。

**前提条件**

Apple JDK 1.6 (Mac OS X 10.5.2 以降) でサポートされている、64 ビットのプロセッサを搭載した Intel ベースの Macintosh コンピュータを使用して、管理ツールを動作させます。http://www.ianywhere.jp/tech/1061806-os_components.html を参照してください。

**内容と備考**

Mac OS X

### ♦ Sybase Central の起動とサンプルデータベースへの接続 (Mac OS X の場合)

1. Finder で、*/Applications/SQLAnywhere* にある [Sybase Central] をダブルクリックします。12
2. **[接続]** » **[SQL Anywhere 12 に接続]** をクリックします。
3. **[アクション]** ドロップダウンリストで、**[ODBC データソースを使用した接続]** をクリックします。
4. **[ODBC データソース名]** をクリックし、その下のボックスに **SQL Anywhere 12 Demo** と入力します。
5. **[接続]** をクリックします。

#### 結果

Sybase Central は、サンプルデータベースに接続されています。

#### 次の手順

データベース内のデータとオブジェクトを表示し、編集できます。

**参照**


- 「[接続] ウィンドウを開く (Sybase Central の場合)」14 ページ
- 「Sybase Central の起動とサンプルデータベースへの接続 (UNIX コマンドラインの場合)」818 ページ
- 「Sybase Central の起動とサンプルデータベースへの接続 (Linux の [アプリケーション] メニューの場合)」819 ページ
- 「Sybase Central の起動 (Windows の場合)」816 ページ


## Sybase Central の起動とサンプルデータベースへの接続 (UNIX コマンドラインの場合)

Sybase Central は、データベースサーバ、データベース、含まれるオブジェクトを管理するため、SQL 文とコマンドラインユーティリティの代わりとして使用できるグラフィックツールです。

#### 前提条件

SQL Anywhere ユーティリティは、事前にソース化されている必要があります。

#### 内容と備考

UNIX コマンドライン

### ♦ Sybase Central の起動とサンプルデータベースへの接続

1. ターミナルセッションで次のコマンドを実行します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
scjview
```

Sybase Central が開きます。

2. **[接続] » [SQL Anywhere 12 に接続]** をクリックします。

3. **[アクション]** ドロップダウンリストで、**[ODBC データソースを使用した接続]** をクリックします。

4. **[ODBC データソース名]** をクリックし、その下のボックスに **SQL Anywhere 12 Demo** と入力します。

5. **[接続]** をクリックします。

### 結果

Sybase Central は、サンプルデータベースに接続されます。

### 次の手順

データベース内のデータとオブジェクトを表示し、編集できます。

### 参照


- 「UNIX および Mac OS X ファイル」444 ページ
- 「[接続] ウィンドウを開く (Sybase Central の場合)」14 ページ
- 「Sybase Central の起動とサンプルデータベースへの接続 (Mac OS X の場合)」817 ページ
- 「Sybase Central の起動とサンプルデータベースへの接続 (Linux の [アプリケーション] メニューの場合)」819 ページ
- 「Sybase Central の起動 (Windows の場合)」816 ページ


## Sybase Central の起動とサンプルデータベースへの接続 (Linux の [アプリケーション] メニューの場合)

Sybase Central は、データベースサーバ、データベース、含まれるオブジェクトを管理するため、SQL 文とコマンドラインユーティリティの代わりとして使用できるグラフィックツールです。

### 前提条件

Linux の [**Applications**] メニューをサポートする Linux バージョン。

メニュー項目は、SQL Anywhere インストール環境の一部としてインストールされます。

### 内容と備考

Linux

**♦ Sybase Central の起動とサンプルデータベースへの接続 (Linux の [アプリケーション] メニューの場合)**

1. **[アプリケーション]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[Sybase Central]** をクリックします。

   Sybase Central が開きます。

2. **[接続]** » **[SQL Anywhere 12 に接続]** をクリックします。

3. **[アクション]** ドロップダウンリストで、**[ODBC データソースを使用した接続]** をクリックします。

4. **[ODBC データソース名]** をクリックし、その下のボックスに **SQL Anywhere 12 Demo** と入力します。

5. **[接続]** をクリックします。

**結果**

Sybase Central は、サンプルデータベースに接続されます。

**次の手順**

データベース内のデータとオブジェクトを表示し、編集できます。

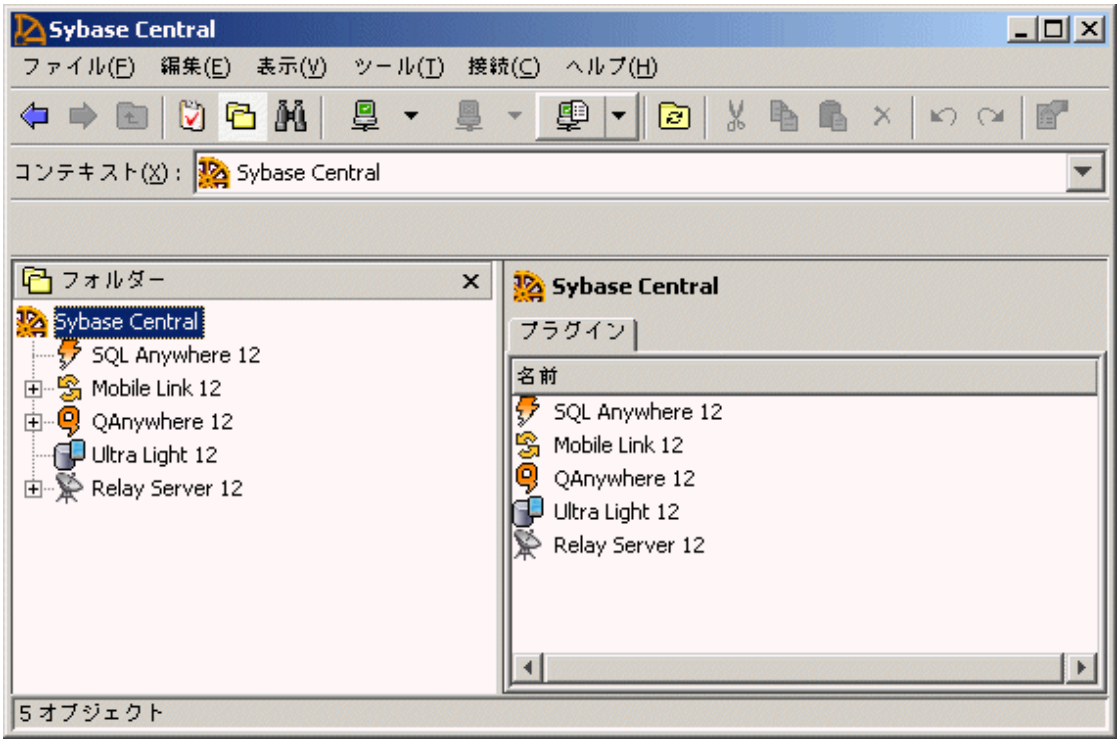
**参照**


- 「[接続] ウィンドウを開く (Sybase Central の場合)」14 ページ
- 「Sybase Central の起動とサンプルデータベースへの接続 (Mac OS X の場合)」817 ページ
- 「Sybase Central の起動とサンプルデータベースへの接続 (UNIX コマンドラインの場合)」818 ページ
- 「Sybase Central の起動 (Windows の場合)」816 ページ


## Sybase Central のナビゲーション

この項では、Sybase Central のユーザーインターフェイスのナビゲーション方法について説明します。

Sybase Central のメインウィンドウ




Sybase Central のウィンドウは左右に並んだ 2 つのウィンドウ枠に分割されています。

### 1 : 左ウィンドウ枠

**[ビュー]** メニューでは、次のどれを左ウィンドウ枠に表示するか選択できます。

- **[フォルダー] ウィンドウ枠**　このウィンドウ枠には、データベースオブジェクトの階層ビューが表示されます。

  **[フォルダー]** のオブジェクトツリーに表示されるのはコンテナのみです。他のオブジェクトのコンテナでないオブジェクトは表示されません。たとえば、左ウィンドウ枠には [カラム] フォルダー (コンテナ) は表示されますが、カラムそのものは項目のため表示されず、代わりに右ウィンドウ枠に表示されます。

- **[タスク] ウィンドウ枠**　このウィンドウ枠には、現在選択されているデータベースオブジェクトのタスクリストが表示されます。

- **[検索] ウィンドウ枠**　このウィンドウ枠では、プラグイン内のオブジェクトを検索できます。「Sybase Central でのデータベースの検索」822 ページを参照してください。

#### ♦ 左ウィンドウ枠のレイアウトの変更

1. Sybase Central を起動します。
2. タスクリストまたはフォルダーリストを表示するか、機能を検索するかに応じて、**[表示]** をクリックしてから **[タスク]**、**[フォルダー]**、または **[検索]** をクリックします。

**2 : 右ウィンドウ枠**

右ウィンドウ枠には、現在選択されているコンテナの内容が表示されます。右ウィンドウ枠には、左ウィンドウ枠で選択したコンテナの内容と、選択したコンテナに関するその他の情報を表示するタブがあります。

**[表示]** » **[カラムの選択]** をクリックすると、右ウィンドウ枠のタブに表示されるカラムを設定できます。

**[ツール]** » **[オプション]** ウィンドウをクリックすると、右ウィンドウ枠の表示方法を変更できます。

一度データベースやデータベースサーバーに接続すると、メインウィンドウでオブジェクトをナビゲーションしたり選択したりして管理できるようになります。

**3 : ツールバー**

メインウィンドウのツールバーには、一般的なコマンドのボタンがあります。ツールバーの表示／非表示を切り替えるには、**[表示]** » **[ツールバー]** » **[標準ツールバー]** をクリックします。メインツールバーでは、次の作業を実行できます。

- オブジェクトフォルダーのナビゲーション
- データベース、データベースサーバー、製品プラグインへの接続または接続の解除
- **[タスク]** ウィンドウ枠、**[フォルダー]** ウィンドウ枠、または **[検索]** ウィンドウ枠の表示
- **[接続プロファイル]** ウィンドウにアクセス (**[ツール]** メニューからも可能)
- 現在のフォルダーのビューの再表示
- オブジェクトの切り取り、コピー、貼り付け、または削除
- 変更の取り消しと再実行
- 選択したオブジェクトのプロパティウィンドウの表示

**4: [コンテキスト] ドロップダウンリスト**

**[コンテキスト]** ドロップダウンリストは、ツールバーの下に表示されるリストで、プラグインのオブジェクトフォルダーをナビゲーションできます。

**5 : ステータスバー**

メニュー間をナビゲーションすると、メインウィンドウの下部に表示されるステータスバーに、メニューコマンドの簡単な説明が表示されます。ステータスバーの表示／非表示を切り替えるには、**[表示]** » **[ステータスバー]** をクリックします。

## Sybase Central でのデータベースの検索

Sybase Central を使用すると、指定したデータベースオブジェクトをデータベース内で検索するか、データベースオブジェクトの SQL 内で文字列を検索できます。

♦ **指定したオブジェクトの検索 (Sybase Central の場合)**

1. **[表示]** » **[検索ウィンドウ枠]** をクリックします。

   左ウィンドウ枠に、**[検索]** ウィンドウ枠が表示されます。

2. 検索のオプションを設定します。

3. **[検索]** をクリックします。

   左ウィンドウ枠の **[結果]** に検索結果が表示されます。

4. 結果をダブルクリックすると、その結果が右ウィンドウ枠に表示されます。

**SQL Anywhere 12** プラグインを使用すると、次のような検索を実行できます。

- **[SQL を検索] (プロシージャー、イベント、ファンクション、トリガー)**　このオプションを選択すると、プロシージャー、イベント、ファンクション、トリガーの SQL 内も検索されます。
- **[動的プロパティ (接続、統計、ロック) の検索]**　このオプションを選択すると、接続しているユーザー、SQL Remote 統計情報、テーブルロック、テーブルページの使用状況などの動的プロパティ内も検索されます。

Mobile Link 12 プラグインを使用すると、次のような検索を実行できます。

- **[スクリプトを検索]**　このオプションを選択すると、同期スクリプト内も検索されます。

## Sybase Central キーボードショートカット

Sybase Central には、以下のキーボードショートカットが用意されています。

| ファンクションキー | 説明 |
|---|---|
| Alt + Enter | 選択した項目のプロパティウィンドウを開きます。 |
| [Ctrl + C] | 選択した内容をクリップボードにコピーします。 |
| [Ctrl + V] | クリップボードの内容を挿入します。 |
| [Ctrl + X] | 選択した内容を切り取ってクリップボードに移動します。 |
| [Delete] | 選択されている内容を削除します。 |
| [F1] | Sybase Central のヘルプを開きます。 |
| [F5] | 選択したフォルダーの内容を再表示します。 |
| [F9] | **[接続プロファイル]** ウィンドウを開きます。 |

| ファンクションキー | 説明 |
|---|---|
| [F11] | 複数のプラグインがロードされている場合は **[接続]** メニューを開きます。ロードされているプラグインが 1 つである場合、[F11] キーを押すとそのプラグインの **[接続]** ウィンドウが開きます。 |
| [F12] | Sybase Central の接続が 1 つだけの場合は切断します。接続が複数の場合は、[F12] キーを押すと **[切断]** ウィンドウが開き、切断する接続を選択できます。 |
| [Shift + F10] | 選択したオブジェクトのポップアップメニューを開きます。 |

## コードエディターの使用

コードエディターは、Sybase Central の右ウィンドウ枠に **[SQL]** タブとして表示されます。Sybase Central に別のウィンドウとして表示されたり、Interactive SQL に **[SQL 文]** ウィンドウ枠として表示されることもあります。

コードエディターには、標準のテキスト編集機能に加えて次のような機能があります。

- ツールバーとステータスバー
- 構文の自動強調表示
- 言語依存インデント
- テキストの検索と置換機能
- ファイルを開いたり保存したりする機能 (この機能を使用できるかどうかは、使用しているプラグインによって決まります)。
- コード印刷機能
- テキスト補完 (コード入力時)

**♦ 別ウィンドウでコードエディターを開く (Sybase Central の場合)**

1. 左ウィンドウ枠で、ストアドプロシージャー、ビュー、トリガーなどのデータベースオブジェクトを選択します。
2. **[ファイル]** » **[新しいウィンドウで編集]** をクリックします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### コードエディターのカスタマイズ

**♦ [SQL] タブの編集中にコードエディターを設定する**

1. **[ファイル]** » **[エディターのカスタマイズ]** をクリックします。
2. 各タブで設定を行います。**[OK]** をクリックします。

**♦ 別のウィンドウで編集中にコードエディターを設定する**

**[オプション]** ウィンドウを使用して、コードエディターの表示をカスタマイズできます。このウィンドウを使用して変更できる設定は、フォアグラウンドとバックグラウンドの色と、コードエディター全体の外観です。ユーザーが加える変更は、次のセッションに移行しても有効です。

1. **[ツール]** » **[オプション]** をクリックします。
2. 各タブで設定を行います。
3. **[OK]** をクリックします。

## コードエディターのキーボードショートカット

Sybase Central には、コードエディター用に次のキーボードショートカットが用意されています。

| ファンクションキー | 説明 |
|---|---|
| [Alt + F4] | コードエディターを閉じます (別のウィンドウで編集中の場合)。または、Sybase Central を閉じます (Sybase Central の右ウィンドウ枠でテキストを編集中の場合)。 |
| [Backspace] | 選択されている内容を削除します。何も選択されていない場合は、[Backspace] キーを押すとカーソルの左にある 1 文字が削除されます。 |
| [Ctrl + ]] | カーソルを閉じカッコまで移動します。カッコ、大カッコ、角カッコ、山カッコに一致させるときに、このショートカットを使用します。 |
| [Ctrl + A] | コードエディターのウィンドウの内容全体を選択します。 |
| [Ctrl + Backspace] | カーソルの左の 1 ワードを削除します。 |
| [Ctrl + C] | 選択したテキストをクリップボードにコピーします。 |
| [Ctrl + Delete] | カーソルの右の 1 ワードを削除します。 |

| ファンクションキー | 説明 |
|---|---|
| [Ctrl + End] | カーソルをコードエディターのウィンドウの一番下に移動します。 |
| [Ctrl + F] | **[検索／置換]** ウィンドウを開きます。現在のウィンドウでテキストを検索していない場合は、指定したテキストを検索して置換できます。それ以外の場合は、指定したテキストの次の出現箇所を検索します。 |
| [Ctrl + F3] | 現在選択されているテキストの次の出現箇所を検索します。 |
| [Ctrl + G] | **[移動]** ウィンドウを開きます。コードエディターのウィンドウ内で移動先の行位置を指定できます。 |
| [Ctrl + Home] | カーソルをコードエディターのウィンドウの先頭に移動します。 |
| [Ctrl + L] | 現在の行を削除します。 |
| [Ctrl + 左矢印] | カーソルを 1 ワード分戻します。 |
| [Ctrl + N] | コードエディターのウィンドウの内容をクリアし、現在のファイルを閉じます (存在する場合)。このショートカットは、Sybase Central の右ウィンドウ枠の **[SQL]** タブからは使用できません。 |
| [Ctrl + O] | コードエディターが別ウィンドウとして開かれている場合に、ファイルを開きます。このショートカットは、Sybase Central の右ウィンドウ枠の **[SQL]** タブからは使用できません。 |
| [Ctrl + P] | コードエディターのウィンドウの内容を印刷します。印刷テキストの外観を設定するには、**[ツール]** » **[オプション]** » **[印刷]** をクリックします。 |
| [Ctrl + 右矢印] | カーソルを 1 ワード分進めます。 |
| [Ctrl + S] | コードエディターのウィンドウの内容を保存します。 |
| [Ctrl + Shift + ]] | 選択範囲を閉じカッコまで拡張します。カッコ、大カッコ、角カッコ、山カッコに一致させるときに、このショートカットを使用します。 |
| [Ctrl + Shift + End] | 選択範囲をコードの終わりまで拡張します。 |
| [Ctrl + Shift + F3] | 現在選択されているテキストより前の出現箇所を検索します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| ファンクションキー | 説明 |
|---|---|
| [Ctrl + Shift + Home] | 選択範囲をコードの先頭まで拡張します。 |
| [Ctrl + Shift + L] | 現在の行を削除します。 |
| [Ctrl + Shift + 左矢印] | 選択範囲を 1 ワード分逆方向に拡張します。 |
| [Ctrl + Shift + 右矢印] | 選択範囲を 1 ワード分先まで拡張します。 |
| [Ctrl + Shift + U] | 選択している内容を大文字に変換します。 |
| [Ctrl + Shift + ピリオド(.)] | コードエディターのウィンドウで選択されたテキストの行インデントを増やします。テキストが選択されていない場合、インデントは現在の行に適用されます。 |
| [Ctrl + Shift + カンマ(,)] | コードエディターのウィンドウで選択されたテキストの行インデントを減らします。テキストが選択されていない場合、インデントは現在の行に適用されます。 |
| [Ctrl + U] | 選択している内容を小文字に変換します。 |
| [Ctrl + V] | クリップボードの内容を現在のカーソル位置に挿入します。 |
| [Ctrl + X] | 選択したテキストを切り取ります。 |
| [Ctrl + Y] | 直前に取り消した操作をやり直します。 |
| [Ctrl + Z] | 最後の操作を取り消します。 |
| [Ctrl + マイナス記号 (-)] | SQL コメントインジケーターの二重ハイフン (--) を追加および削除します。<br><br>既存のテキストをコメントに変換するには、[コードエディター] ウィンドウでテキストを選択し、[Ctrl + マイナス記号] キーを押します。SQL コメントインジケーターの二重ハイフンが、選択したテキストを含む各行の先頭に追加されます。<br><br>テキストが選択されていない場合、コメントインジケーターは現在の行の先頭に追加されます。<br><br>コメントインジケーターを削除するには、テキストを選択して [Ctrl + マイナス記号] キーを押します。<br><br>「コメント」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |

| ファンクションキー | 説明 |
| --- | --- |
| [Ctrl + スラッシュ (/)] | SQL コメントインジケーターの二重スラッシュ (//) を追加および削除します。<br><br>既存のテキストをコメントに変換するには、[コードエディター] ウィンドウでテキストを選択し、[Ctrl + スラッシュ] キーを押します。SQL コメントインジケーターの二重スラッシュが、選択したテキストを含む各行の先頭に追加されます。<br><br>テキストが選択されていない場合、コメントインジケーターは現在の行の先頭に追加されます。<br><br>コメントインジケーターを削除するには、テキストを選択して [Ctrl + スラッシュ] キーを押します。<br><br>「コメント」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| [Delete] | 選択されている内容を削除します。 |
| [下矢印] | カーソルを 1 行下に移動します。 |
| [End] | カーソルを現在の行の末尾に移動します。 |
| [F3] | **[検索／置換]** ウィンドウを開きます。現在のウィンドウでテキストを検索していない場合は、指定したテキストを検索して置換できます。それ以外の場合は、指定したテキストの次の出現箇所を検索します。 |
| [Home] | カーソルを現在の行の行頭、または現在の行のテキストの先頭に移動します。 |
| [左矢印] | カーソルを 1 文字左に移動します。 |
| [Page Down] | カーソルを現在のページの末尾に移動します。 |
| [Page Up] | カーソルを現在のページの先頭に移動します。 |
| [右矢印] | カーソルを 1 文字右に移動します。 |
| [Shift + 下矢印] | 選択範囲を 1 行下まで拡張します。 |
| [Shift + End] | 現在の行を選択します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| ファンクションキー | 説明 |
|---|---|
| [Shift + F3] | **[検索／置換]** ウィンドウを開きます。テキストが選択されていない場合は、指定したテキストを検索して置換できます。テキストが選択されている場合は、そのテキストの前の出現箇所を検索します。 |
| [Shift + F10] | フォーカスのある領域のポップアップメニューを表示します。<br><br>このキーボードショートカットは、領域を右クリックする代わりに使用します。 |
| [Shift + Home] | 選択範囲を現在の行のテキストの先頭まで拡張します。 |
| [Shift + 左矢印] | 選択範囲を、現在選択している文字の左隣の文字まで拡張します。 |
| [Shift + Page Down] | 選択範囲を 1 ページ分下へ拡張します。 |
| [Shift + Page Up] | 選択範囲を 1 ページ分上へ拡張します。 |
| [Shift + 右矢印] | 選択範囲を、現在選択している文字の右隣の文字まで拡張します。 |
| [Shift + 上矢印] | カーソルを 1 行分上へ拡張します。 |
| [上矢印] | カーソルを 1 行上に移動します。 |

## ログビューアー

ログビューアーは Sybase Central のウィンドウで、製品からのメッセージを表示して格納します。次のようなタイプのメッセージを表示します。

- **情報**　現在のセッションに関する基本的な情報メッセージ
- **警告**　発生したアクションに対する警告メッセージ
- **エラー**　失敗したアクションに対するエラーメッセージ

表示するメッセージのタイプや番号を制限したり、特定のプラグインからのメッセージだけを表示したりできます。また、メッセージをファイルに保存したり、すべてのメッセージをリストからクリアしたりすることもできます。

Sybase Central で作業している場合は、**[ツール]** メニューからログビューアーにアクセスできます。

♦ **ログビューアーを開く (Sybase Central の場合)**

1. **[ツール]** » **[ログビューアー]** をクリックします。

   Sybase Central ログビューアーが開き、現在メッセージがあればそれを表示します。

2. **[ビュー]** メニューを使用して、ログに記録するメッセージのタイプを設定します。

## SQL Anywhere 12 プラグイン

**SQL Anywhere 12** プラグインを使用して、既存データベースのアップグレード、新しいデータベースの作成、データベースの管理を行うことができます。モードを選択するには、**[モード]** メニュー、またはモードに対応するツールバーのボタンを使用します。

**SQL Anywhere 12** プラグインは、次のいずれかのモードで操作します。

- **設計モード**　設計モードでは、テーブル、ユーザー、トリガー、インデックス、リモートデータベースサーバーなどのデータベースオブジェクトを作成したり、変更したりできます。また、テーブルへのデータの追加、新しいデータベースの作成、既存データベースのアップグレードを行うことも可能です。

  設計モードで操作中に SQL Anywhere データベースで実行可能なタスクの詳細については、「テーブル、ビュー、インデックス」『SQL Anywhere サーバー SQL の使用法』を参照してください。

- **デバッグモード**　デバッグモードでは、SQL Anywhere のデバッガーを使用して、SQL のストアドプロシージャー、トリガー、イベントハンドラーの開発に役立てることができます。

  「SQL Anywhere のデバッガー」『SQL Anywhere サーバー SQL の使用法』を参照してください。

- **アプリケーションプロファイリングモード**　アプリケーションプロファイリングモードでは、データベースのアプリケーションプロファイリングや診断トレーシングを設定できます。生成されるデータは、アプリケーションがデータベースとやりとりする方法を理解するのに役立ちます。また、パフォーマンスの問題を特定し、解消するためにも有用です。

  「アプリケーションプロファイリング」『SQL Anywhere サーバー SQL の使用法』を参照してください。

### ウィザードによって生成された SQL 文とユーティリティコマンドの表示

ほとんどの **SQL Anywhere** プラグインウィザードの最後のページでは、指定された選択に基づいてウィザードで実行される SQL 文とユーティリティコマンドが表示されます。

**[完了]** をクリックすると、SQL 文かユーティリティコマンドまたはその両方が実行されます。または、SQL 文をクリップボードにコピーし、**[キャンセル]** をクリックしてウィザードを終了した後、Interactive SQL を使用して SQL 文を実行することもできます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

この機能を使用すると、データベースを変更することなくウィザードを使用して SQL スクリプトを生成できます。

**参照**


- 「[接続] ウィンドウを開く (Sybase Central の場合)」14 ページ
- 「テーブル、ビュー、インデックス」『SQL Anywhere サーバー SQL の使用法』
- 「Sybase Central を使用した Windows Mobile データベースの作成」419 ページ


## SQL Anywhere 12 プラグインのデータベースオブジェクト

**SQL Anywhere 12** プラグインでは、既存のデータベースオブジェクトをコピーして、同じデータベース内の別のロケーションか、まったく別のデータベースに挿入できます。

オブジェクトをコピーするには、Sybase Central の左ウィンドウ枠でオブジェクトを選択し、該当するフォルダーまたはコンテナにドラッグします。または、オブジェクトをコピーして、該当するフォルダーまたはコンテナに貼り付けます。新しいオブジェクトが作成され、元のオブジェクトのコードが新しいオブジェクトにコピーされます。同じデータベース内でオブジェクトをコピーする場合は、新しいオブジェクトの名前を変更してください。

ビューのコピーを作成すると、ビュー定義のみが新しい場所にコピーされることに注意してください。パーミッションなど、その他のビューのプロパティはコピーされません。

データベース内の別のオブジェクトにオブジェクトを貼り付けることもできます。たとえば、テーブルをユーザーに貼り付けると、このテーブルに対するパーミッションがこのユーザーに与えられます。

Sybase Central では、次にリストされているオブジェクトのいずれかをコピーすると、そのオブジェクトの SQL がクリップボードにコピーされるため、Interactive SQL やテキストエディターなどの他のアプリケーションに貼り付けることができます。たとえば、Sybase Central でインデッ

クスをコピーし、テキストエディターに貼り付けると、そのインデックスの CREATE INDEX 文が表示されます。**SQL Anywhere 12** プラグインにある次のオブジェクトをコピーできます。

- アーティクル
- 検査制約
- カラム
- DB 領域
- ディレクトリアクセスサーバー
- ドメイン
- イベント
- 外部環境
- 外部ログイン
- 外部キー
- インデックス
- ログインマッピング (統合化ログインと Kerberos ログイン)
- ログインポリシー
- メンテナンスプランレポート
- メンテナンスプラン
- メッセージ型
- Mobile Link ユーザー
- プライマリキー
- プロシージャーとファンクション
- パブリケーション
- リモートサーバー
- スケジュール
- シーケンスジェネレーター
- 空間参照系
- 空間計測単位
- SQL Remote サブスクリプション
- 同期サブスクリプション
- システムトリガー
- テキスト設定オブジェクト
- テキストインデックス
- テーブル
- トリガー
- 一意性制約
- ユーザーとグループ
- ビュー
- Web サービス

## データベーススキーマの比較

Sybase Central を使用して、2 つのデータベーススキーマを比較できます。

### 前提条件

この作業を実行するための前提条件はありません。

### 内容と備考

データベースを比較する際、オブジェクトはオブジェクト ID ではなく、名前、名前と所有者、または名前と所有者の組み合わせで照合されます。

### ♦ データベーススキーマの比較

1. Sybase Central の **SQL Anywhere 12** プラグインを使用すると、比較する 2 つのデータベースに接続できます。
2. **[ツール]** » **[SQL Anywhere 12]** » **[データベースの比較]** をクリックします。

   **[データベースの比較]** ウィンドウが表示されます。
3. 比較する 2 つのデータベースが、**[データベース 1]** フィールドと **[データベース 2]** フィールドに指定されていることを確認します。
4. **[比較]** をクリックします。

   **[Objects]** タブに、データベース間の違いの概要がリスト化されます。
5. **[Objects]** タブのアイテムをクリックすると、そのオブジェクトの SQL 文を参照できます。背景が赤の場合、オブジェクトの SQL は **[データベース 1]** のみにあり、背景が緑の場合、オブジェクトの SQL は **[データベース 2]** のみにあり、背景が青の場合、オブジェクトの SQL は両方のデータベースにあることを示します。

### 結果

データベースが比較されます。

### 次の手順

**[データベース 1]** のスキーマを **[データベース 2]** のスキーマと一致させるために変換したい場合、変換に必要な SQL 文が **[SQL Script]** タブに表示されます。

### 参照

- 「Interactive SQL での SQL スクリプトファイルの実行」『SQL Anywhere サーバー SQL の使用法』
- 「Interactive SQL」840 ページ

## もう一方と一致させるためのデータベーススキーマの変換

Sybase Central を使用して、2 つのデータベーススキーマを比較できます。比較は SQL 文形式で出力されるのでそれを **[データベース 1]** に対して実行すると **[データベース 2]** と同じにすることができます。

### 前提条件

生成された SQL スクリプトには、**[データベース 1]** のスキーマを **[データベース 2]** のスキーマと同じにするために必要な文が含まれます。スクリプトを実行する前に **[データベース 1]** はバッ

クアップすることをおすすめします。また、スクリプトの実行中は、データベースに対する他の接続がないことを確認します。

### 内容と備考

次に示す操作は、この機能を使用して **[データベース 1]** には実行することはできないため、比較中に生成される SQL 文のリストには含まれていません。これらの操作のいずれかが必要な場合は、生成される SQL 文のリストの一番上に、操作を説明する通知が表示されます。

- DB 領域のファイル名の変更
- ドメインの変更
- リモートサーバーの読み込み専用設定の変更
- テーブルの DB 領域の変更
- グローバルテンポラリテーブルの COMMIT アクションか SHARE BY ALL 設定またはその両方の変更
- テーブルまたはグローバルテンポラリテーブルでのカラムの順序の変更
- パブリケーションの同期タイプの変更

**注意**
データベースを比較する際、オブジェクトはオブジェクト ID ではなく、名前、名前と所有者、または名前と所有者の組み合わせで照合されます。**[データベース 1]** にはある **[データベース 2]** にはないのすべてのオブジェクト名 (名前が変更されたテーブルとカラムを含む) は、削除され再作成されるので、その際にデータが失われる場合があります。

### ♦ もう一方と一致させるためのデータベーススキーマの変換

1. Sybase Central の **SQL Anywhere 12** プラグインを使用すると、比較する 2 つのデータベースに接続できます。
2. **[ツール]** » **[SQL Anywhere 12]** » **[データベースの比較]** をクリックします。

   **[データベースの比較]** ウィンドウが表示されます。
3. スキーマを変換するデータベースが **[データベース 1]** フィールドに指定され、スキーマの変換元データベースが **[データベース 2]** フィールドに指定されていることを確認します。
4. **[比較]** をクリックします。

   **[Objects]** タブに、データベース内のオブジェクトがリスト化されます。**[SQL スクリプト]** タブに生成された SQL 文が表示されます。
5. **[Objects]** タブでアイテムをクリックすると、そのオブジェクトの SQL 文の違いを参照できます。背景が赤の場合、オブジェクトの SQL は **[データベース 1]** のみにあり、背景が緑の場

合、オブジェクトの SQL は **[データベース 2]** のみにあり、背景が青の場合、オブジェクトの SQL は両方のデータベースにあることを示します。

6. **[SQL スクリプト]** タブの **[Interactive SQL で開く]** をクリックすると、**[SQL 文]** ウィンドウ枠に文を貼り付けた状態で Interactive SQL を開くことができます。文は必要に応じて編集、実行できます。SQL 文を実行すると **[データベース 1]** を **[データベース 2]** と同じにできます。

**結果**

**[データベース 1]** のスキーマが **[データベース 2]** のスキーマと同じになるよう変換されます。

**次の手順**

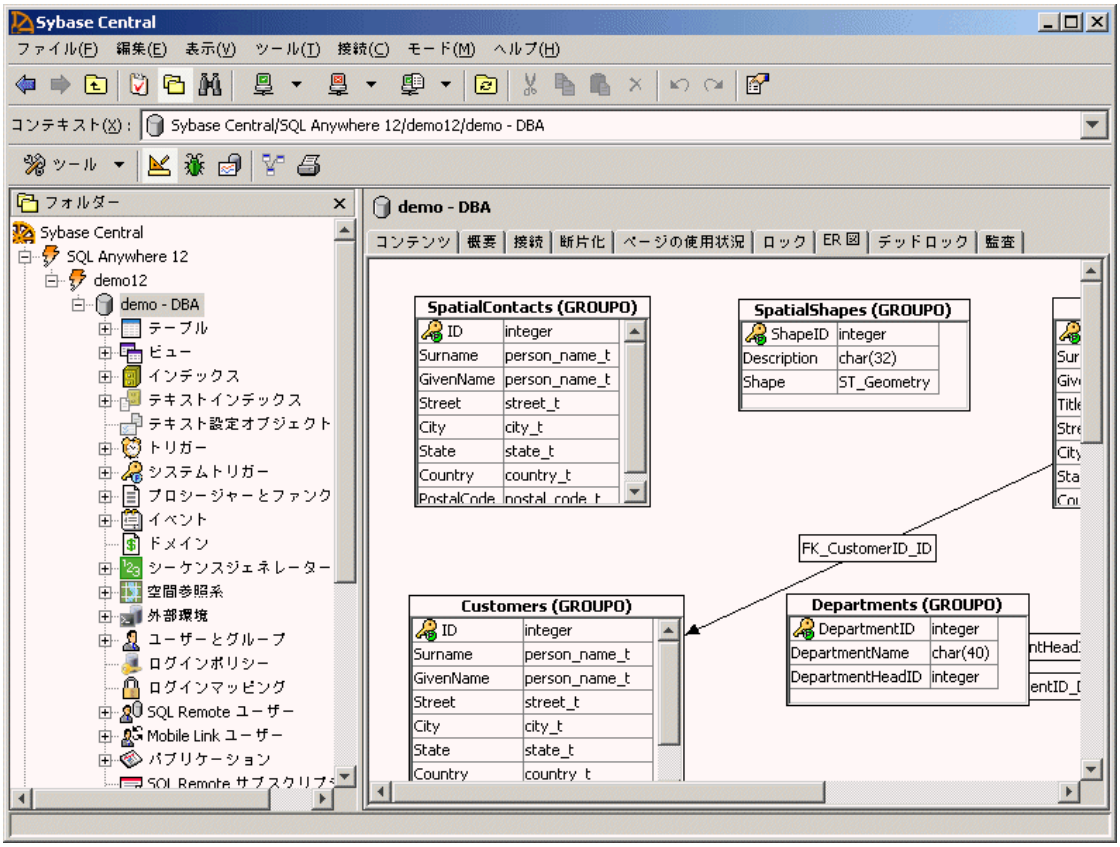
なし。

**参照**

- 「Interactive SQL での SQL スクリプトファイルの実行」『SQL Anywhere サーバー SQL の使用法』
- 「Interactive SQL」840 ページ

## SQL Anywhere 12 プラグインからの ER 図の表示

**SQL Anywhere 12** プラグインからデータベースに接続すると、データベース内のテーブルの ER 図を表示できます。データベースを選択し、右ウィンドウ枠の **[ER 図]** タブをクリックして ER 図を表示します。




ER 図でオブジェクトを並べ替えると、変更は Sybase Central セッション間で保持されます。テーブルをダブルクリックすると、そのテーブルのカラム定義を表示できます。

図のテーブルは、データベースのフィルタリング設定に従って表示されます。フィルタリングは所有者別に行われます。

### ♦ ER 図に含まれるテーブルの変更 (Sybase Central の場合)

1. Sybase Central の左ウィンドウ枠でデータベースを選択し、**[ファイル]** » **[所有者フィルターの設定]** をクリックします。
2. ER 図で参照したいテーブルのデータベースユーザーを選択し、**[OK]** をクリックします。
3. **[ファイル]** » **[所有者別にオブジェクトをフィルター]** をクリックします。
4. 右ウィンドウ枠の **[ER 図]** タブをクリックします。
5. **[ファイル]** » **[ER 図のテーブルを選択]** をクリックします。
6. **[ER 図のテーブルを選択]** ウィンドウで **[追加]** ボタンと **[削除]** ボタンを使用して、**[選択したテーブル]** リストに表示されるテーブルをカスタマイズします。
7. **[OK]** をクリックします。

**参照**

- 「SQL Anywhere データベースの作成」113 ページ

## データベースの正常性と統計情報

設計モードでは、**[概要]** タブにデータベースサーバーの概要と機能が表示されます。このタブには、次の項目があります。

- **[データベース]**　このウィンドウ枠は左上隅にあり、データベースサーバーに関する一般的な情報が表示されます。

  SQL Anywhere データベースサーバーソフトウェアを更新するには、**[更新のチェック]** をクリックします。「ソフトウェアの更新」907 ページを参照してください。

- **[機能]**　このウィンドウ枠は左下隅にあり、データベースと、その製品および機能を視覚的に表示します。図のノードをクリックすると、右側の **[正常性と統計情報]** ウィンドウ枠で対応するセクションが展開し、もう一度ノードをクリックするとセクションが折りたたまれます。

> **注意**
> Mobile Link と QAnywhere の情報の取り出しを開始する必要があります。そうしないと、これらのノードは不明なノードとしてグレー表示されます。[Mobile Link、QAnywhere、Notifier]838 ページを参照してください。

- **[正常性と統計情報]**　このウィンドウ枠は右側にあり、データベースの全体的なステータスに関する統計と情報を表示します。次の折りたたみ可能なウィンドウ枠があります。

  - **[統計情報]**　ディスクとの間で読み書きされたページ数など、一般的な統計情報を表示します。スケジュールされていない要求がある場合は、警告を表示します。詳細を確認するには警告をクリックしてください。

  - **[DB 領域]**　すべての DB 領域を表に示します。DB 領域のサイズがディスクの残り空き領域の 10 % に満たない場合や、DB 領域ファイルが見つからない場合は、警告を表示します。詳細を確認するには警告をクリックしてください。

  - **[トランザクションログ]**　必要に応じてトランザクションログとトランザクションログミラーの情報を表示します。このウィンドウ枠は、データベースにトランザクションログがある場合にのみ表示されます。ログファイルのサイズがディスクの残り空き領域の 10 % に満たない場合は、警告を表示します。詳細を確認するには警告をクリックしてください。

  - **[接続しているユーザー]**　接続しているユーザーとトランザクションの統計情報を表示します。トランザクションがあれば、上位 5 つのトランザクション時間を表に示します。ブロックされた接続があれば、すべてのブロックされた接続を表に示します。ブロックされた接続がある場合は、警告を表示します。詳細を確認するには警告をクリックしてください。

○ **[データベースのミラーリング]**　システムのプライマリサーバーに接続しているときに、プライマリサーバー、監視サーバー、ミラーサーバー、コピーサーバーの情報とデータベースミラーリングシステムまたは読み込み専用のスケールアウトシステムの情報を表示します。このウィンドウ枠は、ミラーリングまたはスケールアウトを使用している場合にのみ表示されます。監視サーバーまたはミラーサーバーが切断されている場合は、警告を表示します。詳細を確認するには警告をクリックしてください。

○ **[リモートサーバー]**　データベースが使用するリモートサーバーを表に示します。このウィンドウ枠は、リモートサーバーが存在する場合にのみ表示されます。リモートサーバーが切断されている場合や、リモートサーバーが接続を確立できない場合は、警告を表示します。詳細を確認するには警告をクリックしてください。

> **注意**
> JDBC リモートサーバーでは、JDBC リモートサーバーがプロキシオブジェクトへのアクセスを試行した場合のみ、接続が切断されたか失われたことを示す警告が表示されます。「JDBC クラスの設定上の注意」『SQL Anywhere サーバー SQL の使用法』を参照してください。

○ **[Mobile Link、QAnywhere、Notifier]**　Mobile Link、QAnywhere、および Notifier の統計値を表示します。このウィンドウ枠は、Mobile Link のテーブルとビューがデータベース内にある場合のみ表示されます。

> **注意**
> このウィンドウ枠の情報を再表示するには、**[再表示]** ボタンをクリックする必要があります。他のウィンドウ枠の情報と異なり、このウィンドウ枠の情報は、**[表示]** » **[再表示]** をクリック (またはツールバーの **[再表示]** アイコンをクリック) しても再表示されません。再表示はデータベースのパフォーマンスに影響を与えることがあるため、この情報はユーザーが個別に再表示する必要があります。

○ **[SQL Remote ユーザー]**　すべての SQL Remote ユーザーと、それぞれの最後の送信時刻および受信時刻を表に示します。このウィンドウ枠は、データベースに SQL Remote ユーザーが存在する場合にのみ表示されます。

## データベースのドキュメント化

**データベースドキュメントウィザード**を使用して、SQL Anywhere データベース内のオブジェクトに関するドキュメントを生成できます。生成されるドキュメントには、次のデータベースオブジェクトに関する情報が含まれます。

- イベント
- ファンクション
- プロシージャー
- テーブル
- トリガー
- ビュー


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

オブジェクトの定義に加えて、ドキュメントは各オブジェクトの依存性と参照も示します。たとえば、プロシージャー *dbo.sa_migrate_data* のドキュメントには、更新、挿入、削除対象のテーブル、また呼び出し元のプロシージャー名が含まれます。オブジェクトコメントやシステムプロシージャーをドキュメントに含めることもできます。

生成されるドキュメントは、ナビゲーションや確認がしやすいように HTML ファイルに保存されます。このドキュメントは、システムのドキュメント化と確認に役立ちます。

**♦ データベースドキュメントの生成 (Sybase Central の場合)**

1. **SQL Anywhere 12** プラグインを使用して、ドキュメントを生成するデータベースに接続します。
2. **[ツール]** » **[SQL Anywhere 12]** » **[データベースドキュメントの生成]** をクリックします。
3. データベースドキュメントウィザードの指示に従います。

## Sybase Central の左ウィンドウ枠の設定

**♦ タイプフィルターの設定 (Sybase Central の場合)**

Sybase Central の左ウィンドウ枠に表示するデータベースオブジェクトを指定できます。

1. データベースに接続します。
2. 左ウィンドウ枠でデータベースを右クリックし、**[タイプフィルターの設定]** をクリックします。
3. Sybase Central の左ウィンドウ枠に表示するオブジェクトのタイプを選択します。
4. (任意) **[これを新しいデータベースのデフォルトのタイプフィルターにする]** をクリックすると、デフォルトのタイプフィルターが設定したフィルターに設定されます。デフォルトのタイプフィルターは、新しいデータベースなど、タイプフィルターが指定されていないデータベースに接続するときに使用されます。
5. **[OK]** をクリックします。

## データベース変更のロギング

SQL Anywhere プラグインによって実行された、データベースを変更するすべての SQL 文を *.sql* ファイルに記録できます。

**参照**

- 「Sybase Central での SQL 文のロギング」154 ページ

# Interactive SQL

Interactive SQL は SQL Anywhere に付属しているツールです。このツールを使用すると、SQL Anywhere と Ultra Light のデータベースに対して SQL 文の実行、スクリプトのビルド、データベースのデータ表示を実行できます。Interactive SQL は、次の目的で使用できます。

- SQL 文をデータベースサーバーに送信する。「Interactive SQL での SQL 文の実行」848 ページを参照してください。
- データベース内の情報をブラウズする。「Interactive SQL での SQL 文の実行」848 ページを参照してください。
- 結果セットにあるデータを編集する。「Interactive SQL での結果セット」859 ページを参照してください。
- データベースにデータを読み込む。「インポートウィザード (Interactive SQL) でのデータのインポート」『SQL Anywhere サーバー SQL の使用法』を参照してください。
- クエリ結果をファイルまたは別のデータベースにエクスポートする。「クエリ結果のエクスポート」『SQL Anywhere サーバー SQL の使用法』を参照してください。
- スクリプトファイルを実行する。「Interactive SQL での SQL スクリプトファイルの実行」『SQL Anywhere サーバー SQL の使用法』を参照してください。
- インデックスコンサルタント (クエリパフォーマンスの向上に役立つツール) を実行する。「インデックスコンサルタント」『SQL Anywhere サーバー SQL の使用法』を参照してください。
- クエリエディター (あらゆる種類のクエリの設計、分析、テストに役立つツール) にアクセスする。「Interactive SQL のクエリエディター」864 ページを参照してください。
- クエリの実行プランを表示する。「Interactive SQL のプランビューアーとグラフィカルなプラン」867 ページを参照してください。
- SVG などのイメージを表示する。「Interactive SQL とグラフィックイメージ」869 ページを参照してください。
- 空間データを表示する。「空間データのイメージとしての表示 (Interactive SQL の場合)」『SQL Anywhere サーバー 空間データサポート』を参照してください。

Interactive SQL は、Windows、Solaris、Linux、Mac OS X で利用できます。http://www.ianywhere.jp/tech/1061806-os_components.html を参照してください。

**注意**
管理ツールは、Apple JDK 1.6 (Mac OS X 10.5.2 以降) でサポートされている、64 ビットのプロセッサーを搭載した Intel ベースの Macintosh コンピューターだけで動作します。http://www.ianywhere.jp/tech/1061806-os_components.html を参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### Interactive SQL のみから使用される SQL 文

Interactive SQL は、SQL Anywhere と Ultra Light のデータベースでサポートされるすべての SQL 文と、Interactive SQL でのみ使用可能な SQL 文をサポートします。「Interactive SQL の SQL 文」878 ページを参照してください。

## Interactive SQL の起動

Interactive SQL は、コマンドプロンプト、Windows の **[スタート]** メニュー、Sybase Central 内から起動できます。Interactive SQL を使用すると、SQL Anywhere データベース、Sybase IQ データベース、Ultra Light データベース、または汎用 ODBC データベースに接続できます。

**♦ Interactive SQL の起動 (コマンドラインの場合)**

● 次のコマンドを実行します。

```
dbisql
```

データベースへの接続パラメーターを指定する -c オプションを省略するなど、接続パラメーターの指定が不十分であると、**[接続]** ウィンドウが表示されます。このウィンドウで、データベースへの接続情報を入力できます。「接続パラメーター」320 ページを参照してください。

Interactive SQL を起動し、サンプルデータベースに接続するには、次のコマンドを実行します。

```
dbisql -c "UID=DBA;PWD=sql;DSN=SQL Anywhere 12 Demo"
```

サポートされるオプションの詳細については、「Interactive SQL ユーティリティ (dbisql)」975 ページを参照してください。

**♦ Interactive SQL の起動 (Windows の場合)**

1. **[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[Interactive SQL]** をクリックします。
2. **[データベースタイプを変更]** をクリックして、接続するデータベースのタイプを選択します。SQL Anywhere データベース、Sybase IQ データベース、Ultra Light データベース、汎用 ODBC データベースに接続できます。

   たとえば、SQL Anywhere データベースに接続するには、**[データベースタイプを変更]** をクリックし、**[SQL Anywhere]** をクリックします。
3. **[接続]** ウィンドウで、データベースの接続情報を入力します。

   たとえば、SQL Anywhere サンプルデータベースに接続するには、次の手順に従います。

   a. **[認証]** フィールドで **[データベース]** をクリックします。

   b. **[アクション]** ドロップダウンリストで、**[ODBC データソースを使用した接続]** をクリックします。

c. **[ODBC データソース名]** をクリックし、その下のボックスに **SQL Anywhere 12 Demo** と入力します。

4. **[接続]** をクリックします。

### ♦ Interactive SQL の起動 (Sybase Central の場合)

1. **[ツール]** » **[SQL Anywhere 12]** » **[Interactive SQL を開く]** をクリックします。
2. **[接続]** ウィンドウで、データベースの接続情報を入力します。

   たとえば、SQL Anywhere サンプルデータベースに接続するには、次の手順に従います。

   a. **[認証]** フィールドで **[データベース]** をクリックします。

   b. **[アクション]** ドロップダウンリストで、**[ODBC データソースを使用した接続]** をクリックします。

   c. **[ODBC データソース名]** をクリックし、その下のボックスに **SQL Anywhere 12 Demo** と入力します。

3. **[接続]** をクリックします。

**ヒント**
次のいずれかの方法で Sybase Central から Interactive SQL にアクセスすることもできます。

●データベースを選択し、**[ファイル]** » **[Interactive SQL を開く]** をクリックする。

●データベースを右クリックし、**[Interactive SQL を開く]** をクリックする。

●ストアドプロシージャーを右クリックし、**[Interactive SQL から実行]** をクリックする。**[SQL 文]** ウィンドウ枠にプロシージャーへの呼び出しが指定されて Interactive SQL が開き、ストアドプロシージャーが実行されます。

●テーブルまたはビューを右クリックし、**[Interactive SQL によるデータ表示]** をクリックする。SELECT * FROM table-name が指定されて Interactive SQL が開き、クエリが実行されます。

### ♦ Interactive SQL の起動 (UNIX コマンドラインの場合)

**注意**
以降の手順は、SQL Anywhere ユーティリティのソースを指定済みであることを前提としています。「Unix および Mac OS X 環境変数」443 ページを参照してください。

1. ターミナルセッションで次のコマンドを実行します。

```
dbisql
```

2. **[接続]** ウィンドウで、データベースの接続情報を入力します。
3. **[接続]** をクリックします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

♦ **Interactive SQL の起動 (Mac OS X の場合)**

1. Finder で、*/Applications/SQLAnywhere* にある [Interactive SQL] をダブルクリックします。12
2. **[接続]** ウィンドウで、データベースの接続情報を入力します。
3. **[接続]** をクリックします。

♦ **Interactive SQL の起動 (Linux デスクトップアイコンの場合)**

Linux のデスクトップアイコンをサポートする Linux バージョンを使用していて、SQL Anywhere のインストール時にこれらのアイコンをインストールするように選択した場合は、次の手順を使用できます。

1. **[アプリケーション]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[Interactive SQL]** をクリックします。
2. **[接続]** ウィンドウで、データベースの接続情報を入力します。
3. **[接続]** をクリックします。

**参照**

- 「[接続] ウィンドウを開く (Sybase Central の場合)」14 ページ

## Interactive SQL のナビゲーション

Interactive SQL のウィンドウは、次のウィンドウ枠に分かれています。

- **[SQL 文]**　このウィンドウ枠には、データのアクセスや変更に使用する SQL 文を入力します。

  **[SQL 文]** ウィンドウ枠の左側のカラムに表示される行番号を使用して、次のことができます。

  - **1 行の選択**　1 行を選択するには、行番号をクリックします。または、その行にカーソルを置いて、[Ctrl + カンマ (,)] キーを押します。
  - **複数行の選択**　複数行を選択するには、開始行をクリックして終了行までドラッグします。
  - **文の選択**　1 行をダブルクリックして、その行に対応する SQL 文全体を選択します。または、その文にカーソルを置いて、[Ctrl + ピリオド (.)] キーを押します。

    「Interactive SQL のキーボードショートカット」878 ページを参照してください。

- **[結果]**　**[結果]** ウィンドウ枠には、**[結果]** と **[メッセージ]** の 2 つのタブがあります。これらのタブは、**[結果]** ウィンドウ枠の下部に表示されます。

**[結果]** タブには、実行した文の結果が表示されます。たとえば、データベースから特定のデータを検索する SQL 文を使用すると、**[結果]** タブに、上のウィンドウ枠の探索条件と一致するカラムとローが表示されます。**[結果]** タブに表示された結果セットは編集できます。「Interactive SQL での結果セット」859 ページを参照してください。

**[メッセージ]** タブには、Interactive SQL で実行する SQL 文に関するデータベースサーバーのメッセージが表示されます。

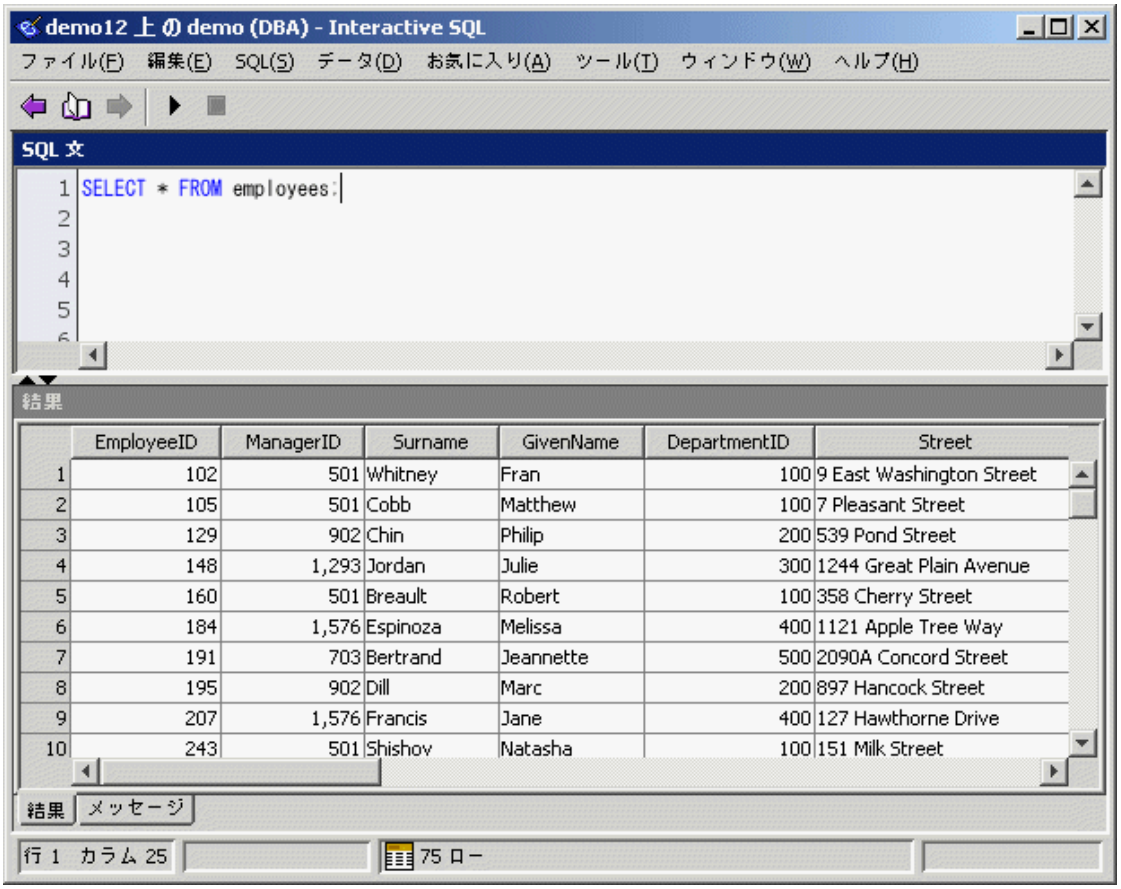
SQL Anywhere データベースのグラフィカルなプランの結果と、Ultra Light データベースのテキストプランの結果は、それぞれ別の [プランビューアー] ウィンドウに表示されます。詳細とノードの統計を含むグラフィカルなプランを作成する867 ページを参照してください。

Interactive SQL からデータベースに接続しているとき、タイトルバーには次のような接続情報が表示されます。

server-name 上の database-name ( userid )

たとえば、SQL Anywhere 12 Demo ODBC を使用してサンプルデータベースに接続すると、タイトルバーには次のような情報が含まれます。

demo12 上の demo ( DBA )




 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## Interactive SQL のウィンドウ

Interactive SQL のすべてのウィンドウは、**[ツール]** メニュー、**[データ]** メニュー、**[お気に入り]** メニューを使用して表示できます。これらのウィンドウを使用して、Interactive SQL の設定、クエリに挿入するテーブル名やプロシージャー名の検索、クエリの編集、結果セットのエクスポート、ファイルや接続情報のお気に入りとしての保存を行うことができます。

**[ツール]** メニューから表示できるウィンドウは次のとおりです。

- **[テーブル名のルックアップ]**　**[テーブル名のルックアップ]** ウィンドウでは、テーブル名やカラム名を検索して、それを **[SQL 文]** ウィンドウ枠に挿入できます。「Interactive SQL でのテーブル、カラム、プロシージャーの検索」857 ページを参照してください。

- **[プロシージャー名のルックアップ]**　**[プロシージャー名のルックアップ]** ウィンドウでは、プロシージャー名を検索して、それを **[SQL 文]** ウィンドウ枠に挿入できます。「Interactive SQL でのテーブル、カラム、プロシージャーの検索」857 ページを参照してください。

- **[クエリの編集]**　クエリエディターによって、Interactive SQL で SELECT 文の作成と編集をグラフィック表示で行う方法が提供されます。「Interactive SQL のクエリエディター」864 ページを参照してください。

- **[インデックスコンサルタント]**　インデックスコンサルタントは、適切なインデックスを選択する手助けをします。インデックスコンサルタントを使えば、特定のクエリに対するさまざまなインデックスのメリットを分析できます。「クエリに対するインデックスコンサルタントの推奨内容の確認」『SQL Anywhere サーバー SQL の使用法』を参照してください。

- **[プランビューアー]**　**プランビューアー**は、SQL Anywhere データベースのグラフィカルなプランと、Ultra Light データベースのテキストプランを表示するためのグラフィカルツールです。「Interactive SQL のプランビューアーとグラフィカルなプラン」867 ページを参照してください。

- **[空間ビューアー]**　**空間ビューアー**は、空間データを表示するためのグラフィカルツールです。「空間データのイメージとしての表示 (Interactive SQL の場合)」『SQL Anywhere サーバー 空間データサポート』を参照してください。

- **[オプション]**　**[オプション]** ウィンドウでは、Interactive SQL のコマンド、外観、データのインポートとエクスポート、メッセージなどのオプションを設定できます。

**[データ]** メニューから表示できるウィンドウは次のとおりです。

- **[エクスポート]**　**エクスポートウィザード**が開き、結果セットをエクスポートできます。「エクスポートウィザードを使用したデータのエクスポート」『SQL Anywhere サーバー SQL の使用法』を参照してください。

- **[インポート]**　**インポートウィザード**が開き、ファイルまたはデータベースからデータをインポートできます。「インポートウィザード (Interactive SQL) でのデータのインポート」『SQL Anywhere サーバー SQL の使用法』を参照してください。

**[お気に入り]** メニューから表示できるウィンドウは次のとおりです。

- **[お気に入りに追加]**　このウィンドウでは、SQL ファイル、SQL 文、接続情報をお気に入りとして保存できます。「SQL スクリプトファイルと接続のお気に入りリストへの保存 」854 ページを参照してください。

- **[お気に入りの整理]**　このウィンドウでは、お気に入りの管理や整理ができます。「SQL スクリプトファイルと接続のお気に入りリストへの保存 」854 ページを参照してください。

- **[お気に入りのインポート]**　このウィンドウでは、お気に入りを *.fav* ファイルからインポートできます。「お気に入りの SQL スクリプトファイルと接続の共有」856 ページを参照してください。

- **[お気に入りのエクスポート]**　このウィンドウでは、お気に入りを *.fav* ファイルにエクスポートできます。「お気に入りの SQL スクリプトファイルと接続の共有」856 ページを参照してください。

- **[お気に入りを表示]**　Interactive SQL ウィンドウの左側に **[お気に入り]** ウィンドウが表示されます。「SQL スクリプトファイルと接続のお気に入りリストへの保存 」854 ページを参照してください。

## Interactive SQL のカスタマイズ

Interactive SQL では、**[オプション]** ウィンドウを使用してタブやウィンドウ枠の設定を変更できます。Interactive SQL を配備するときに、**[オプション]** ウィンドウで表示または有効化する機能を制御できます。「管理ツールの設定」『SQL Anywhere サーバー プログラミング』を参照してください。

### ♦ Interactive SQL のカスタマイズ

1. Interactive SQL で、**[ツール]** » **[オプション]** をクリックします。
2. 左ウィンドウ枠でオプションをクリックし、必要なオプションを指定します。
3. **[OK]** をクリックします。

### ♦ Interactive SQL での結果セットの表示方法を設定する

1. 結果をスクロール可能なテーブルとして表示するか、またはテキストとして表示するかを選択します。

   たとえば、**[データ]** をクリックし、次のいずれかを選択します。

   - **[結果をスクロール可能なテーブルとして表示]**　このフォーマットの結果セットは編集できます。これはデフォルトです。
   - **[結果をテキストとして表示]**　このオプションを選択すると、結果セットは等幅フォントのテキストで表示されます。このフォーマットの結果セットは編集できません。

2. クエリを実行します。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

テーブル編集の変更を反映させるには、新しいクエリを実行する必要があります。

3. 結果セット内を右クリックし、次のいずれかを選択します。
   - **[カラムのサイズをデータに合わせる]** テーブルカラムの幅が、カラム内の値がトランケートされないように広げられます。
   - **[カラムのサイズをウィンドウに合わせる]** すべてのカラムが水平スクロールバーのないウィンドウ内に収まるように、テーブルカラムは同じ幅になります。

### 警告メッセージの無効化

Interactive SQL に表示される警告メッセージの一部を無効にすることができます。たとえば、**[SQL 文]** ウィンドウ枠でテキストを保存しないで **[ファイル]** » **[終了]** をクリックした場合に表示される警告を非表示にすることができます。

**♦ 警告メッセージの無効化**

1. **[ツール]** » **[オプション]** » **[メッセージ]** をクリックします。
2. **[オプションのメッセージ]** リストに表示されているメッセージをクリアします。

**参照**

- 「データを表示した場合の予期しない記号のトラブルシューティング」479 ページ

## Interactive SQL でのファイルの操作

Windows プラットフォームでは、Interactive SQL を *.sql* スクリプトファイルのデフォルトエディターにできます。

**♦ Interactive SQL を *.sql* ファイルのデフォルトエディターに設定する (Windows の場合)**

1. Interactive SQL で、**[ツール]** » **[オプション]** をクリックします。
2. 左ウィンドウ枠で、**[一般]** をクリックします。
3. **[Interactive SQL を .SQL ファイルとプランファイルのデフォルトエディターにする]** をクリックします。
4. **[OK]** をクリックします。

**参照**

- 「SQL スクリプトファイル」『SQL Anywhere サーバー SQL の使用法』
- 「Interactive SQL での SQL スクリプトファイルの実行」『SQL Anywhere サーバー SQL の使用法』

## Interactive SQL での *.sql* ファイルのバックアップ

*.sql* ファイルを編集すると、次の場合に Interactive SQL によって自動的にバックアップコピーが作成されます。

- 文を実行した場合
- ファイルが 30 秒間変更されなかった場合

ファイルの編集中に Interactive SQL が予期せず終了した場合は、次回 Interactive SQL を起動するときに、ファイルをリカバリするよう要求されます。

お気に入りをバックアップすることもできます。

**参照**

- 「お気に入りの SQL スクリプトファイルと接続の共有」856 ページ

## Interactive SQL での SQL 文の実行

Interactive SQL の主な使用目的の 1 つは、テーブルデータをブラウズすることです。Interactive SQL では、データベースサーバーに要求を送信することによって情報を検索します。一方、データベースサーバーは、情報を調べ、それを Interactive SQL に返します。

SELECT 文を実行すると、**[結果]** ウィンドウ枠の **[結果]** タブに結果セットが表示されます。デフォルトでは、ロー番号が結果セットの左に表示されます。

**注意**
Interactive SQL でテーブルを表示すると、テーブルを変更しなくても、テーブルに対するスキーマロックがデータベースサーバーによって作成されます。

ただし、結果セットが表示されるときに作成されるデータベーススキーマロックを解放するように Interactive SQL を設定できます。そのように設定するには、Interactive SQL で **[ツール]** » **[オプション]** » **[SQL Anywhere]** をクリックし、**[データベースロックの自動解放]** をクリックします。

このオプションを選択すると、結果セットを返す文を実行した後、データベースにコミットされていない変更が接続にあるかどうかが Interactive SQL によって確認されます。コミットされていない変更がなかった場合は Interactive SQL によってスキーマロックが解放されます。それ以外の場合、スキーマロックは解放されません。つまり、データベースにコミットされていない変更がある場合、スキーマロックは解放されません。

**♦ すべての SQL 文を実行する**

1. **[SQL 文]** ウィンドウ枠にクエリを入力します。
2. [F5] キーを押すか、**[SQL]** » **[実行]** をクリックして、文を実行します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**♦ 選択した SQL 文を実行する**

1. **[SQL 文]** ウィンドウ枠にクエリを入力し、そのクエリを選択します。
2. [F9] キーを押すか、**[SQL]** » **[選択の実行]** をクリックして、文を実行します。

SQL 文を個別に実行するには (デバッグ時など)、**[SQL]** » **[シングルステップ]** をクリックします。**[シングルステップ]** を使用すると、指定された文の実行が完了してから、次に実行する文が選択されます。次の文を実行するには、もう一度 **[シングルステップ]** を実行します。

**♦ SQL 文を 1 文ずつ実行する**

1. **[SQL 文]** ウィンドウ枠にクエリ (複数可) を入力します。
2. 実行する文にカーソルを置きます。
3. **[SQL]** » **[シングルステップ]** をクリックするか、[Shift + F9] キーを押して、指定した文を実行します。

   その SQL 文が実行されると、次の SQL 文が選択されます。
4. 選択された SQL 文を実行するには、[Shift + F9] キーを押します。
5. 実行する選択された文がなくなるまで、前述の手順を繰り返します。

**[実行]** ボタンをクリックして、**[SQL 文]** ウィンドウ枠にある文を実行することもできます。このボタンを押したときにすべての SQL 文を実行するか、選択した文のみを実行するかを設定できます。

**♦ [実行] ツールバーボタンの設定**

1. **[ツール]** » **[オプション]** をクリックします。
2. **[ツールバー]** をクリックします。

   すべての SQL 文を実行する場合は、**[実行]** をクリックします。これがデフォルト設定です。

   選択した SQL 文のみを実行する場合は、**[選択の実行]** をクリックします。

**参照**

- 「管理ツールの設定」『SQL Anywhere サーバー プログラミング』

## Interactive SQL でのコメントの挿入

コメントは、SQL 文または文ブロックに説明テキストを付加するために使用します。データベースサーバーは、コメントを実行しません。SQL Anywhere でサポートされるコメントのタイプは、-- (二重ハイフン)、// (二重スラッシュ)、/* ... */ (スラッシュとアスタリスク) です。「コメント」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

♦ **コメントインジケーターの追加または削除**

- 既存のテキストをコメントに変換するには、まず **[SQL 文]** ウィンドウ枠でテキストを選択します。次に、コメントインジケーターとして二重ハイフンを追加する場合は [Ctrl + マイナス記号 (-)] キー、二重スラッシュを追加する場合は [Ctrl + スラッシュ (/)] キーを押します。SQL コメントインジケーターが、選択したテキストの各行の先頭に追加されます。

  テキストが選択されていない場合、コメントインジケーターは現在の行の先頭に追加されます。

  コメントインジケーターを削除するには、テキストを選択して、[Ctrl + マイナス記号 (-)] キー (コメントインジケーターが二重ハイフンの場合) または [Ctrl + スラッシュ (/)] キー (コメントインジケーターが二重スラッシュの場合) を押します。

### Interactive SQL での SQL 文のインデント

♦ **SQL 文のインデントを変更する**

1. **[SQL 文]** ウィンドウ枠で、インデントを設定するテキストを選択します。テキストが選択されていない場合、インデントは現在の行に適用されます。
2. [Ctrl + ピリオド (.)] キーを押すと、インデントが追加または増加されます。

   [Ctrl + カンマ (,)] キーを押すと、インデントが削除または減少されます。

♦ **インデントするスペースの数を変更する**

1. **[ツール]** » **[オプション]** » **[エディター]** » **[タブ]** をクリックします。
2. **[インデントサイズ]** フィールドに、新しい数を入力します。

## Interactive SQL での複数の SQL 文の実行

各文がコマンドデリミターで終了している間は、Interactive SQL から複数の SQL 文を実行できます。コマンドデリミターは command_delimiter オプションで設定されるもので、デフォルトではセミコロン (;) です。セミコロンを使用する代わりに、セパレーター **go** を独立した行の先頭に配置することもできます。「command_delimiter オプション [Interactive SQL]」887 ページを参照してください。

> **ヒント**
> [F9] キーを押すと、**[SQL 文]** ウィンドウ枠で選択したテキストだけが実行されます。
>
> [Shift + F9] キーを押すと、**[SQL 文]** ウィンドウ枠で選択した文だけが実行され、次に実行する文が選択されます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 複数の結果セットの表示

デフォルトでは、Interactive SQL は、最後に実行された文の最初の結果セットを表示します。Interactive SQL が複数の結果セットを返すことができるようにする方法については、「プロシージャーから複数の結果セットを返す」『SQL Anywhere サーバー SQL の使用法』を参照してください。

### 参照

● 「データを表示した場合の予期しない記号のトラブルシューティング」479 ページ

## Interactive SQL の SQL スクリプトファイル

SQL スクリプトファイルは、SQL 文を含むテキストファイルであり、同じ SQL 文を繰り返し実行する場合に便利です。Interactive SQL を使用して、スクリプトファイルを開いたり、表示、実行、保存を行ったりすることができます。

SQL スクリプトファイルを Interactive SQL から実行するには、次の方法があります。

● Interactive SQL の READ 文を使用して、スクリプトファイルを実行する。たとえば、次の文はファイル *temp.sql* を実行します。

```
READ temp.sql;
```

● スクリプトファイルを **[SQL 文]** ウィンドウ枠にロードして、そこから直接実行する。

スクリプトファイルを **[SQL 文]** ウィンドウ枠にロードするには、**[ファイル]** » **[開く]** をクリックします。メッセージが表示されたら、ファイル名 (たとえば *temp.sql*) を入力します。

● **[ファイル]** » **[スクリプトの実行]** をクリックして、スクリプトファイルをロードしないで実行する。

● スクリプトファイルを Interactive SQL のコマンドライン引数として指定する。

### 参照

● 「Interactive SQL での SQL スクリプトファイルの実行」『SQL Anywhere サーバー SQL の使用法』

## Interactive SQL での COMMIT 文と ROLLBACK 文の実行

COMMIT 文は次の方法で実行できます。

● **[SQL]** » **[コミット]** をクリックします。

● [Ctrl + Shift + C] キーを押す。

● **[SQL 文]** ウィンドウ枠で **Commit** と入力して文を実行する。

ROLLBACK 文は次の方法で実行できます。

- **[SQL]** » **[ロールバック]** を選択します。
- [Ctrl + Shift + R] キーを押す。
- **[SQL 文]** ウィンドウ枠で Rollback と入力して文を実行する。

> **注意**
> **[SQL]** メニューまたはキーボードショートカットを使用して COMMIT または ROLLBACK を実行した場合、**[SQL 文]** ウィンドウ枠の内容は変更されません。ただし、**[結果]** ウィンドウ枠の **[結果]** タブの内容はクリアされます。

**参照**

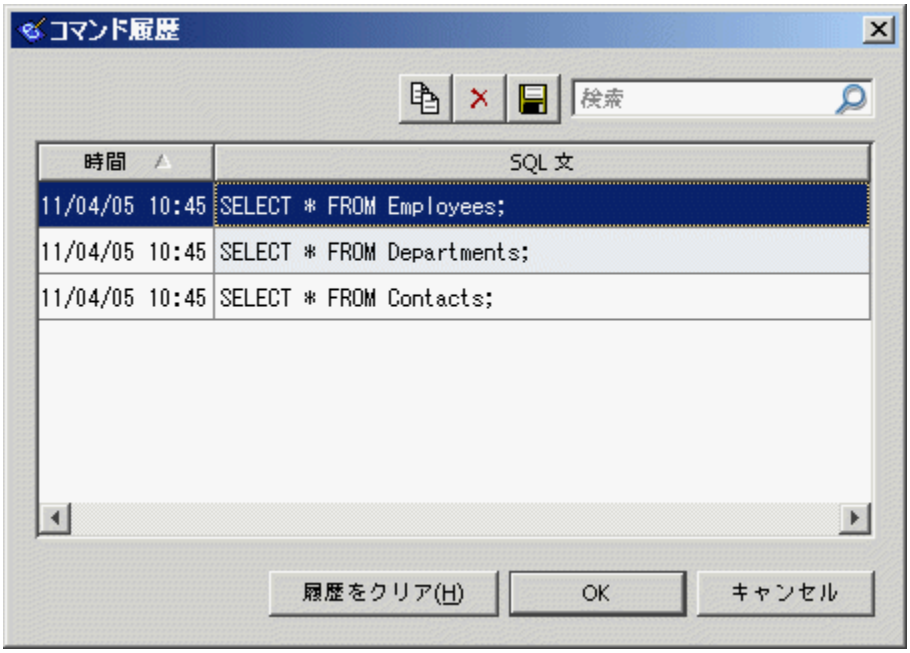

- 「auto_commit オプション [Interactive SQL]」885 ページ
- 「commit_on_exit オプション [Interactive SQL]」888 ページ
- 「COMMIT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「ROLLBACK 文」『SQL Anywhere サーバー SQL リファレンス』
- 「Interactive SQL オプション」884 ページ


## Interactive SQL での文の再呼び出し

文を実行すると、Interactive SQL ではその文が自動的に履歴リストに保存され、次の Interactive SQL セッションまで保持されます。Interactive SQL は、最近使用した文を 50 個まで記録します。

**[コマンド履歴]** ウィンドウで、文のリスト全体を表示できます。**[コマンド履歴]** ウィンドウにアクセスするには、[Ctrl + H] キーを押すか、ツールバーの **[過去の SQL 文のリストを開く]** ボタンをクリックします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1




最後に使用した文はリストの一番下に表示されます。文を再度呼び出すには、その文を選択して **[OK]** をクリックします。コマンドが Interactive SQL の **[SQL 文]** ウィンドウ枠に表示されます。**[コマンド履歴]** ウィンドウから複数の文を選択できます。

また、**[コマンド履歴]** ウィンドウを使用しないで文を再度呼び出すこともできます。ツールバーの **[前の SQL 文の再呼び出し]** と **[次の SQL 文の再呼び出し]** のアイコンを使用するか、[Alt + 右矢印] キーと [Alt + 左矢印] キーのどちらかを押して、文を前後にスクロールします。

**注意**

パスワード情報を含む SQL 文 (CREATE USER、GRANT REMOTE DBA、CONNECT、または CREATE EXTERNLOGIN) を実行する場合は、現在の Interactive SQL セッションの間、**[コマンド履歴]** ウィンドウにパスワード情報が表示されます。

文の履歴がその後の Interactive SQL セッションで表示されるときには、パスワード情報を含む文ではパスワードの代わりに「...」が表示されます。たとえば、Interactive SQL で次の文を実行したと仮定します。

```
CREATE USER testuser
 IDENTIFIED BY testpassword;
```

この場合、以降の Interactive SQL セッションでは **[コマンド履歴]** ウィンドウに次の文が表示されます。

```
CREATE USER testuser
 IDENTIFIED BY ...;
```

### [コマンド履歴] ウィンドウからの文のコピー

♦ **[コマンド履歴] ウィンドウから文をコピーする**

**[コマンド履歴]** ウィンドウから文をコピーして、他の場所で使用できます。複数の文をコピーすると、各文が文デリミター (デフォルトではセミコロン) で区切られます。

1. **[コマンド履歴]** ウィンドウを開きます。
2. 1 つまたは複数の文を選択し、[Ctrl + C] キーを押すか、**[コピー]** をクリックします。
3. **[OK]** をクリックして、選択した文を Interactive SQL の **[SQL 文]** ウィンドウ枠にコピーします。

### [コマンド履歴] ウィンドウからの文の保存

文は、以降の Interactive SQL セッションで使用できるように、テキストファイルに保存することもできます。

♦ **文の履歴をファイルに保存する**

1. **[コマンド履歴]** ウィンドウを開きます。
2. **[.SQL ファイルとして履歴を保存]** ボタンをクリックするか、[Ctrl + S] キーを押します。
3. **[名前を付けて保存]** ウィンドウで、ファイルの保存場所と名前を指定します。

   文の履歴ファイルの拡張子は、*.sql* にします。
4. 設定が完了したら、**[保存]** をクリックします。

SQL スクリプトファイルを **[お気に入り]** リストに保存することもできます。「SQL スクリプトファイルと接続のお気に入りリストへの保存 」854 ページを参照してください。

### [コマンド履歴] ウィンドウからの文の削除

**[コマンド履歴]** ウィンドウの内容は、Interactive SQL セッション間で保持されます。文は、次のいずれかの方法で履歴から削除できます。

- 1 つまたは複数の文を選択し、**[削除]** ボタンをクリックするか [Delete] キーを押して、選択した文をウィンドウから削除します。この操作は取り消せません。
- **[履歴をクリア]** をクリックすると、すべての文がウィンドウから削除されます。この操作は取り消せません。

## SQL スクリプトファイルと接続のお気に入りリストへの保存

Interactive SQL では、頻繁に使用する SQL スクリプトファイルと接続を、お気に入りリストに保存することができます。お気に入りリストは各ユーザーに固有のもので、別のユーザーからは見えません。

### ♦ *.sql* ファイル、SQL 文、または接続をお気に入りに追加する

1. お気に入りに追加する SQL スクリプトファイルを開きます。
2. **[お気に入り]** » **[お気に入りに追加]** をクリックします。
3. **[開いているファイル '*filename*' を追加]** をクリックします。**[名前]** フィールドに *.sql* ファイルの名前を入力します。
4. **[OK]** をクリックします。

### ♦ SQL 文をお気に入りに追加する

1. **[SQL 文]** ウィンドウ枠で、お気に入りに追加する SQL 文を入力します。[SQL 文] ウィンドウ枠の内容は 16384 文字以内にする必要があります。
2. **[お気に入り]** » **[お気に入りに追加]** をクリックします。
3. **[SQL 文の追加]** をクリックします。**[名前]** フィールドにお気に入りの名前を入力します。
4. **[OK]** をクリックします。

### ♦ 接続をお気に入りに追加する

1. データベースに接続します。
2. **[お気に入り]** » **[お気に入りに追加]** をクリックします。
3. **[接続パスワードの保存]** をクリックします。**[名前]** フィールドに接続の名前を入力します。
4. **[OK]** をクリックします。

### ♦ お気に入りをサイドバーに表示する

● **[お気に入り]** » **[お気に入りを表示]** をクリックします。

Interactive SQL ウィンドウの左側に、**[お気に入り]** ウィンドウ枠が表示されます。

### ♦ お気に入りを開く

● **[お気に入り]** をクリックして、開きたいお気に入りを選択します。

### ♦ お気に入りを編集する

1. **[お気に入り]** » **[お気に入りを表示]** をクリックします。

   Interactive SQL ウィンドウの左側に、**[お気に入り]** ウィンドウ枠が表示されます。
2. お気に入りを選択し、右クリックして [編集] をクリックします。

3. ウィンドウの指示に従います。
4. **[保存]** をクリックします。

**参照**

- 「管理ツールの設定」『SQL Anywhere サーバー プログラミング』

### お気に入りの SQL スクリプトファイルと接続の共有

すべてのお気に入りを *.fav* ファイルにエクスポートできます。その後、お気に入りを別のコンピューターにインポートしたり、ファイルをバックアップとして保存したりできます。

**♦ お気に入りをエクスポートする (Interactive SQL の場合)**

1. **[お気に入り]** » **[お気に入りのエクスポート]** をクリックします。
2. *.fav* ファイルの名前を指定し、**[エクスポート]** をクリックします。

**♦ お気に入りをインポートする (Interactive SQL の場合)**

1. **[お気に入り]** » **[お気に入りのインポート]** をクリックします。
2. *filename.fav* ファイルを参照し、**[インポート]** をクリックします。

## Interactive SQL での文のロギング

Interactive SQL のロギング機能を使うと、SQL 文を実行しながら記録できます。Interactive SQL は、ユーザーがロギングプロセスを停止するまで、または現在のセッションを終了するまで、文の記録を続けます。記録された文は、再び使用できるようにログファイルに保存されます。

ロギングを開始すると、正常に動作しなかったものも含めて実行した SQL 文がすべて記録されます。

**♦ 文のロギングを開始する (Interactive SQL の場合)**

1. **[SQL]** » **[ロギングの開始]** をクリックします。
2. **[名前を付けて保存]** ウィンドウで、ログファイルの保存場所と名前を指定します。たとえば、*mylogs.sql* のようなファイル名を指定します。
3. 設定が完了したら、**[保存]** をクリックします。

**♦ 文のロギングを開始する (SQL の場合)**

- START LOGGING 文を実行します。

  たとえば、*c:¥mylogs.sql* という名前のファイルへのロギングを開始するには、次の文を実行します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
START LOGGING 'c:¥mylogs.sql';
```

「START LOGGING 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

**♦ 文のロギングを停止する (Interactive SQL の場合)**

● **[SQL]** » **[ロギングの停止]** をクリックします。

**♦ 文のロギングを停止する (SQL の場合)**

● 次の文を実行します。

```
STOP LOGGING;
```

「STOP LOGGING 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

**ヒント**

●Sybase Central を使用している場合も、実行する文のログを記録できます。「Sybase Central での SQL 文のロギング」154 ページを参照してください。

●さらに、ロギングと実行時間の両方のレポートが有効になっている場合は、ログファイルに実行時間を記録できます。「isql_command_timing オプション [Interactive SQL]」892 ページを参照してください。

## Interactive SQL での文のキャンセル

キャンセル操作によって現在の処理が停止し、次のコマンドのプロンプトが表示されます。Interactive SQL ツールバーで **[停止]** ボタンをクリックすると、文をキャンセルできます。

SQL スクリプトファイルが処理中の場合は、必要なアクション (**コマンドファイルを停止**するか、**続行**するか、**Interactive SQL を終了**する) を指定するためのプロンプトが表示されます。これらのアクションは、Interactive SQL の on_error オプションを使用して制御できます。

**参照**

● 「on_error オプション [Interactive SQL]」898 ページ

## Interactive SQL でのテーブル、カラム、プロシージャーの検索

Interactive SQL に文を入力するときに、現在のデータベースに格納されているテーブル、カラム、またはプロシージャーの名前を検索して、カーソル位置に挿入できます。

### ♦ データベース内のテーブルの名前を検索する

1. **[ツール]** » **[テーブル名のルックアップ]** をクリックするか、[F7] キーを押します。
2. テーブルを検索し、選択します。
3. **[OK]** をクリックして、テーブル名を **[SQL 文]** ウィンドウ枠の現在のカーソル位置に挿入します。

### ♦ データベース内のカラムの名前を検索する

1. **[ツール]** » **[テーブル名のルックアップ]** をクリックするか、[F7] キーを押します。
2. カラムを含むテーブルを検索し選択します。
3. **[カラムを表示]** をクリックします。
4. カラムを選択し、**[OK]** をクリックして、カラム名を **[SQL 文]** ウィンドウ枠の現在のカーソル位置に挿入します。

### ♦ データベース内のプロシージャーの名前を検索する

1. **[ツール]** » **[プロシージャー名のルックアップ]** をクリックするか、[F8] キーを押します。
2. プロシージャーを検索し、選択します。
3. **[OK]** をクリックして、プロシージャー名を **[SQL 文]** ウィンドウ枠の現在のカーソル位置に挿入します。

**[テーブル名のルックアップ]** と **[プロシージャー名のルックアップ]** の各ウィンドウ枠で、検索するテーブルまたはプロシージャーの最初の数文字を入力すると、入力した文字で始まる項目だけを含むようにリストが限定されます。

SQL のワイルドカード文字 '%' (パーセント記号) と '_' (アンダースコア) を使用すると、検索対象を絞り込むことができます。%' は、0 文字以上の任意の文字列を表し、'_' は、任意の 1 文字を表します。

たとえば、profile という語を含むすべてのテーブルをリストするには、**%profile%** と入力します。

テーブル名に含まれるパーセント記号またはアンダースコアを検索する場合は、パーセント記号またはアンダースコアの前にエスケープ文字を付ける必要があります。SQL Anywhere ODBC ドライバーと SQL Anywhere JDBC ドライバーを使用している場合は '~' (チルダ) をエスケープ文字として使用します。

**ヒント**
**[SQL 文]** ウィンドウ枠に入力するとき、Interactive SQL によってデータベースオブジェクト名のテキスト補完がサポートされます。この機能は、テーブルやプロシージャーの名前を検索するときの代替手段として使用できます。「テキスト補完の使用」901 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## Interactive SQL での結果セットからの SQL 文の生成

結果セット内で選択したローに対する、INSERT 文、DELETE 文、UPDATE 文を作成できます。

**♦ Interactive SQL の結果セットから SQL 文を生成する**

1. 文を生成するローを選択します。
2. 選択したローを右クリックし、**[生成]** をクリックしてから、**[INSERT 文]**、**[DELETE 文]**、または **[UPDATE 文]** をクリックします。

   文がクリップボードにコピーされます。

## Interactive SQL での結果セット

Interactive SQL でクエリを実行した後、結果セットのソートや編集をしてデータベースを修正できます。また、結果セットからローを選択し、他のアプリケーションで使用できるようにコピーすることもできます。結果セットのフィールドデリミター、引用文字、エスケープ文字は、それぞれ isql_field_separator、isql_quote、isql_escape_character オプションによって制御されます。これらのオプションは、Interactive SQL の **[オプション]** ウィンドウを使用するか、または SET OPTION 文を実行して表示および変更できます。

Interactive SQL は、ローの編集、挿入、削除をサポートしています。結果セットの編集には、UPDATE、INSERT、DELETE の各文を実行した場合と同様の効果があります。結果セットを編集すると、対応する INSERT 文、UPDATE 文、DELETE 文が Interactive SQL のコマンド履歴に追加されます。

結果セット内のローまたは値を編集するには、値を修正するテーブルまたはカラムに対する適切なパーミッションが必要です。たとえば、ローを削除する場合は、そのローが属しているテーブルに対する DELETE パーミッションが必要です。

次の場合には、結果セットを編集できません。

- プライマリキーを持つテーブルからカラムを選択したが、一部のプライマリキーカラムを選択していない
- JOIN の結果セットを編集しようとした (たとえば、結果セットに複数のテーブルのデータがある場合)
- 編集が無効になっているテーブルを編集しようとした

次の場合には、結果セットの編集に失敗することがあります。

- パーミッションのないローやカラムを編集しようとした
- 無効な値を入力した (たとえば、数値カラムに文字列を入力したり、NULL を使用できないカラムに NULL を入力した場合)

編集に失敗すると、エラーを説明する Interactive SQL エラーメッセージが表示されます。データベーステーブルの値は変更されません。

**参照**


- 「Interactive SQL オプション」884 ページ
- 「Interactive SQL での文の再呼び出し」852 ページ
- 「テーブル編集の有効化と無効化」861 ページ


## Interactive SQL の結果セットからテーブル値を編集する

**♦ 結果セット内のローの編集**

Interactive SQL から、データベーステーブルに存在するローの一部またはすべての値を変更できます。ただし、変更するカラムに対する UPDATE パーミッションがある場合に限ります。また、SQL Anywhere と Ultra Light のデータベースでは、テーブル編集を無効にできません。

結果セットを編集する場合、一度に変更できるのは 1 つのローの値だけです。

1. Interactive SQL でクエリを実行します。
2. **[結果]** タブで、変更したい値をクリックします。
3. 値を右クリックし、**[ローの編集]** をクリックするか、[F2] キーを押します。

   その値を含むテーブルセルに **[...]** ボタンが表示されます。
4. **[...]** をクリックし、次のいずれかのオプションを選択します。
   - **[ウィンドウで編集]**　新しい値 (文字データフィールドにのみ適用されます) を入力できるウィンドウを開きます。
   - **[NULL に設定]**　セルの値を NULL に設定します。カラムが NULL 入力不可の場合、このメニュー項目は表示されません。
   - **[デフォルトに設定]**　セルの値をデフォルトに設定します。このメニュー項目が表示されるのは、新しいローをテーブルに追加するときに、カラムにデフォルト値が定義されている場合のみです。
   - **[ファイルからロード]**　ファイル名を入力できるファイルブラウザーを開き、セルの内容をファイルからロードします。
5. 新しい値を入力します。そのローの他の値を変更する場合は、[Tab] または [Shift + Tab] キーを押して他の値に移動します。
6. ローの値を編集したら、[Enter] キーを押してデータベースを更新します。

   [Esc] キーを押すと、選択した値に対して行った変更をキャンセルできます。
7. COMMIT 文を実行し、テーブルに対する変更を永続的なものにします。たとえば、[Ctrl + Shift + C] キーを押します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### テーブル編集の有効化と無効化

Interactive SQL の **[オプション]** ウィンドウを使用して、テーブル編集を無効にできます。

Interactive SQL を配備するときに、テーブル編集を無効にすることもできます。

**♦ テーブル編集を無効にする (Interactive SQL の場合)**

1. **[ツール]** » **[オプション]** をクリックしてから、**[SQL Anywhere]** または **[Ultra Light]** のいずれかをクリックします。
2. **[スクロール可能なテーブル]** が選択されていることを確認し、**[編集の無効化]** をクリックします。
3. **[OK]** をクリックします。
4. クエリを実行します。

   テーブル編集の変更を反映させるには、新しいクエリを実行する必要があります。

**参照**

- 「管理ツールの設定」『SQL Anywhere サーバー プログラミング』

## Interactive SQL の結果セットからデータベースにローを挿入する

Interactive SQL では、テーブルに新しいローを追加できます。結果セット内のカラム間をタブで移動し、ローに新しい値を追加します。

**♦ 結果セットに新しいローを挿入する**

新しいローを追加するには、テーブルに対する INSERT パーミッションが必要です。

1. 結果セットを右クリックし、**[ローの追加]** をクリックします。

   新しい空白のローが表示され、そのローの最初の値に点滅するカーソルが表示されます。
2. 新しい値を入力し、[Tab] キーを押して次のカラムに移動します。

   カラムには、無効なデータ型は入力できません。たとえば、INT データ型を受け入れるカラムには、文字列は入力できません。

   すべてのカラム値が追加されるまで、この手順を繰り返します。
3. [Enter] キーを押してデータベースを更新します。

### デフォルト値を持つカラムへの値の挿入

デフォルト値を持つカラムに値を追加するとき、セルのエディターには **(デフォルト)** 項目を含むリストが表示されます。デフォルト値を挿入する場合は、**[(デフォルト)]** をクリックします。

同様に、NULL 値を受け入れるカラムである場合、リストには **[(NULL)]** が表示されます。NULL 値を受け入れず、デフォルト値も持たないカラムである場合は、値を入力する必要があります。

### 計算カラムへの値の挿入

結果セット内に計算カラムがあり、計算カラムに値を指定しない場合、値はデータベースが更新されたときに計算されます。しかし、計算カラムに値を指定した場合は、指定した値でデータベースが更新され、値は計算カラムに対して計算されません。

### INPUT 文を使用した新しいローの挿入

Interactive SQL の結果セットから新しいローを挿入するには、INPUT 文で PROMPT 句を使用してローを追加するという方法もあります。PROMPT 句を指定すると、Interactive SQL によって、テーブルの各カラムの値を入力するよう要求されます。たとえば、Products テーブルに新しいローを追加し、各カラムの値を入力するよう求めるメッセージを表示するには、Interactive SQL で次の文を実行します。

```
INPUT INTO Products PROMPT;
```

### 参照


- 「INPUT 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「input_format オプション [Interactive SQL]」890 ページ


## Interactive SQL を使用してデータベースからローを削除する

**♦ 結果セットからローを削除する**

ローを削除するには、テーブルに対する DELETE パーミッションが必要です。

1. 削除するローを選択します。ローを選択するには、次の手順に従います。

   - [Shift] キーを押しながらローをクリックします。
   - [Shift + 上矢印] または [Shift + 下矢印] キーを押します。

   ローが連続していない場合は、それぞれ個別に削除します。

2. [Delete] キーを押します。

   選択したローがデータベーステーブルから削除されます。

3. COMMIT 文を実行し、変更を永続的なものにします。

## Interactive SQL の結果セットからカラム、ロー、セルをコピーする

Interactive SQL に表示される結果セットからセル、ロー、カラムを直接コピーし、他のアプリケーションに貼り付けることができます。ローやカラムをコピーすると、カラム見出しとテーブルデータもクリップボードにコピーされます。一度にコピーできるカラムは 1 つだけです。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

コピーされたデータは、次の Interactive SQL オプションに従ってフォーマットされます。

- 「isql_field_separator オプション [Interactive SQL]」
- 「isql_escape_character オプション [Interactive SQL]」
- 「isql_quote オプション [Interactive SQL]」

これらのオプションは、Interactive SQL で **[オプション]** » **[インポート／エクスポート]** をクリックして変更することもできます。

これらのオプションがデフォルトに設定されている場合、コピーしたデータはカンマで区切られ、文字列は一重引用符で囲まれます。

### ♦ Interactive SQL の結果セットからローをコピーする

1. 1 つのローをコピーするには、ロー内のセルを右クリックし、**[データのコピー]** » **[ロー]** をクリックします。

   複数のローをコピーするには、[Ctrl] キーを押しながらロー内のセルをクリックし、右クリックして **[データのコピー]** » **[ロー]** をクリックします。

   選択したローが、カラム見出しを含めてクリップボードにコピーされます。

2. そのローを、他のアプリケーションに貼り付けることができます。

### ♦ Interactive SQL の結果セットからカラムをコピーする

1. 1 つのカラムをコピーするには、カラム内のセルを右クリックし、**[データのコピー]** » **[カラム]** をクリックします。

   複数のカラムをコピーするには、[Ctrl] キーを押しながらカラム内のセルをクリックし、右クリックして **[データのコピー]** » **[カラム]** をクリックします。

   **[結果]** ウィンドウ枠に結果セット全体が含まれていない場合、選択する前に残りの結果をフェッチするよう要求されます。残りをフェッチしないと、その時点でフェッチされている結果のみが選択されます。

   カラムが、カラム見出しを含めてクリップボードにコピーされます。

2. そのカラムを、他のアプリケーションに貼り付けることができます。

### ♦ Interactive SQL の結果セットからセルをコピーする

1. コピーするセルを右クリックして、**[データのコピー]** » **[セル]** をクリックします。複数のセルをコピーするには、[Ctrl] キーを押しながらセルをクリックし、右クリックして **[データのコピー]** » **[セル]** をクリックします。

   この操作を行った場合、カラム見出しはコピーされず、データのみがクリップボードにコピーされます。引用もされません。

2. セルの内容を、他のアプリケーションに貼り付けることができます。

### Interactive SQL の結果セットのカラムをソートする

♦ **結果セットのカラムをソートする**

● **[結果]** タブのカラムヘッダーをクリックします。そのカラムを基準にして結果がソートされます。

**[結果]** タブに結果セット全体が含まれていない場合、残りの結果をフェッチするよう要求されます。残りをフェッチしないと、現在フェッチされている結果のみがソートされます。

## Interactive SQL のクエリエディター

クエリエディターは、SELECT 文の構築を支援する Interactive SQL のツールです。クエリエディターで SQL クエリを作成したり、それらの SQL クエリをインポートして編集したりできます。クエリが完成したら、**[OK]** をクリックし、クエリを Sybase Central または Interactive SQL にエクスポートして処理します。

♦ **クエリエディターを使用してクエリを作成する**

1. Interactive SQL からデータベースに接続します。
2. クエリエディターを開きます。

   **[ツール]** » **[クエリの編集]** をクリックします。

   SQL 文を Interactive SQL で選択していた場合は、選択した文がクエリエディターに自動的にインポートされます。
3. クエリを作成します。

   クエリエディターを使用しているときはいつでも、ウィンドウの下部の **[SQL]** をクリックすると、構築しているクエリの SQL 文を確認できます。ここでコードを直接編集でき、クエリエディターのユーザーインターフェイスのフィールドが自動的に更新されます。
4. **[OK]** をクリックして、そのクエリを Interactive SQL の **[SQL 文]** ウィンドウ枠に書き込みます。

#### クエリエディターによる SQL 文の使用

クエリエディターでは、クエリを作成するために SQL 文を使用する必要はありません。ただし、次のように、クエリエディターで SQL 文を使用する場合もあります。

●Interactive SQL の **[SQL 文]** ウィンドウ枠でクエリを作成して、コードを強調表示してからクエリエディターを開くと、クエリエディターにクエリをインポートできます。

●クエリエディターを使用しているときはいつでも、ウィンドウの下部の **[SQL]** をクリックすると、構築しているクエリの SQL 文を確認できます。ここでコードを直接編集でき、クエリエディターのフィールドが自動的に更新されます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### クエリエディターの設定

Interactive SQL または Sybase Central でクエリエディターを設定して、SQL が完全なフォーマットで作成される (すべてのテーブルとカラムの名前が完全に修飾され、名前が引用符で囲まれる) ように指定できます。このような特別なフォーマットは通常は必要ありませんが、これによってあらゆる状況において SQL の動作が保証されます。起動時にテーブルのリストを取得することもできます。

Interactive SQL からクエリエディターを設定するには、**[ツール]** » **[オプション]** » **[SQL Anywhere]** » **[クエリエディター]** タブをクリックします。

### [クエリエディター] タブ

クエリエディターには SQL クエリのコンポーネントを設定するための一連のタブがありますが、そのほとんどはオプションです。タブは SQL クエリが通常構築される順序で表示されています。

| [Tab] | 説明 |
|---|---|
| **[テーブル]** タブ | クエリでテーブルを指定するには、このタブを使用します。 |
| **[ジョイン]** タブ | クエリに複数のテーブルを含める場合、テーブルのデータを結合するためのジョイン方式を指定するには、このタブを使用します。ジョイン方式を指定しない場合は、クエリエディターから 1 つの方式が提案されます。テーブルの間に外部キー関係がある場合は、その関係に基づくジョイン条件が生成されます。それ以外の場合は外積 (ベクトル積) が提案されます。クエリを開くとき、クエリエディターはユーザーが指定したジョイン方式をそのまま受け入れます (SQL Anywhere の場合とは異なり、指定のない JOIN が、デフォルトで KEY JOIN になることはありません)。 |
| **[カラム]** タブ | 結果セットのカラムを指定するには、このタブを使用します。カラムを指定しないと、すべてのカラムが表示されます。 |
| **[INTO]** タブ | 結果を変数に割り当てるには、このタブを使用します。 |
| **[WHERE]** タブ | 結果セットのローを制限する条件を指定するには、このタブを使用します。 |
| **[GROUP BY]** タブ | 結果セットのローをグループ化するには、このタブを使用します。 |

| [Tab] | 説明 |
|---|---|
| **[HAVING]** タブ | グループの値に基づいて結果セットのローを制限するには、このタブを使用します。 |
| **[ORDER BY]** タブ | ローをソートするには、このタブを使用します。 |

### クエリエディターのツール

クエリエディターには、次のツールもあります。

| ウィンドウ | 説明 |
|---|---|
| 式エディター | 探索条件の構築や、計算カラムの定義には、式エディターを使用します。「Interactive SQL の式エディター」866 ページを参照してください。 |
| 派生テーブル | メインのクエリエディターとほとんど同じこのウィンドウを使用して、派生テーブルまたはサブクエリを作成します。 |

クエリエディターの各コンポーネントには、コンテキスト別のオンラインヘルプがあります。これには、タブの使用方法の説明や、関連する概念や使用方法を示す SQL Anywhere マニュアルへのリンクが含まれています。

### クエリエディターの制限事項

クエリエディターは、SQL Anywhere の SELECT 文を構築します。ビューを作成するようには設計されていませんが、Interactive SQL でビューを作成してからクエリエディターで参照することはできます。また、SELECT 文以外の、UPDATE 文などの SQL 文の作成にも対応していません。作成されるのは単独の SELECT 文であり、複数の SELECT 文の UNION 演算や INTERSECT 演算は構築されません。さらに、クエリエディターでは Transact-SQL 構文はサポートされません。

### 参照

- 「クエリ」『SQL Anywhere サーバー SQL の使用法』
- 「SELECT 文」『SQL Anywhere サーバー SQL リファレンス』

## Interactive SQL の式エディター

式エディターによって、探索条件、計算カラム、サブクエリを作成できます。既存の式を編集する場合は、その式を強調表示してから式エディターを開いてください。強調表示をしないと、**[OK]** をクリックしたときに、式エディターで作成した式は既存の式の後ろに追加されます。

### 項目

- **[式]**　このボックスで式を構築します。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

- **[カラム]**　このボックスには、クエリのカラムがリストされます。式にカラムを挿入するには、ここでカラムをダブルクリックするか、**[式]** ウィンドウ枠にカラムを直接入力します。
- **[関数]**　関数は、データベースの情報を返すために使用される定義済みの式です。関数を式に挿入するには、ドロップダウンリストから式を選択します。「SQL 関数」『SQL Anywhere サーバー SQL リファレンス』を参照してください。
- **[ストアドプロシージャー]**　このボックスには、使用できるストアドプロシージャーがリストされます。
- **[数字キーパッド]**　数字キーパッドは、ウィンドウの左下にあります。クリックした数字が式に挿入されます。キーボードから数字を入力することもできます。
- **[比較演算子]**　= などの算術記号は、ウィンドウの下部中央に表示されます。記号をクリックすると式に挿入されます。ここに表示されるのは一部の一般的な演算子だけですが、その他の演算子はキーボードから入力できます。「比較演算子」『SQL Anywhere サーバー SQL リファレンス』を参照してください。
- **[論理演算子]**　AND などの論理演算子は、ウィンドウの右下に表示されます。演算子をクリックすると式に挿入されます。これらは一般的に使用される論理演算子のサブセットです。キーボードから入力すればあらゆる論理演算子を使用できます。「論理演算子」『SQL Anywhere サーバー SQL リファレンス』を参照してください。
- **[サブクエリを編集し、エディターに結果を挿入] ボタン**　このボタンは、論理演算子と同じ場所にあります。このボタンをクリックすると、サブクエリを作成できるウィンドウが表示されます。

### サブクエリの追加

サブクエリを作成するには、式エディターを開き、**[サブクエリを編集し、エディターに結果を挿入]** ボタン (**[NOT]** ボタンの横) をクリックします。これで **[サブクエリ]** ウィンドウが開きます。このウィンドウは、タイトルバーに **[サブクエリ]** と表示されること以外はメインのクエリエディターと同じです。サブクエリはメインクエリの場合と同じ方法で作成します。

### 参照

- 「Interactive SQL のクエリエディター」864 ページ
- 「探索条件」『SQL Anywhere サーバー SQL リファレンス』
- 「探索条件内のサブクエリ」『SQL Anywhere サーバー SQL リファレンス』

## Interactive SQL のプランビューアーとグラフィカルなプラン

プランビューアーは、SQL Anywhere データベースのグラフィカルなプランと、Ultra Light データベースのテキストプランを表示するためのグラフィカルツールです。「Ultra Light の実行プランの表示」『Ultra Light データベース管理とリファレンス』を参照してください。

**♦ 詳細とノードの統計を含むグラフィカルなプランを作成する**

1. Interactive SQL を開き、データベースに接続します。

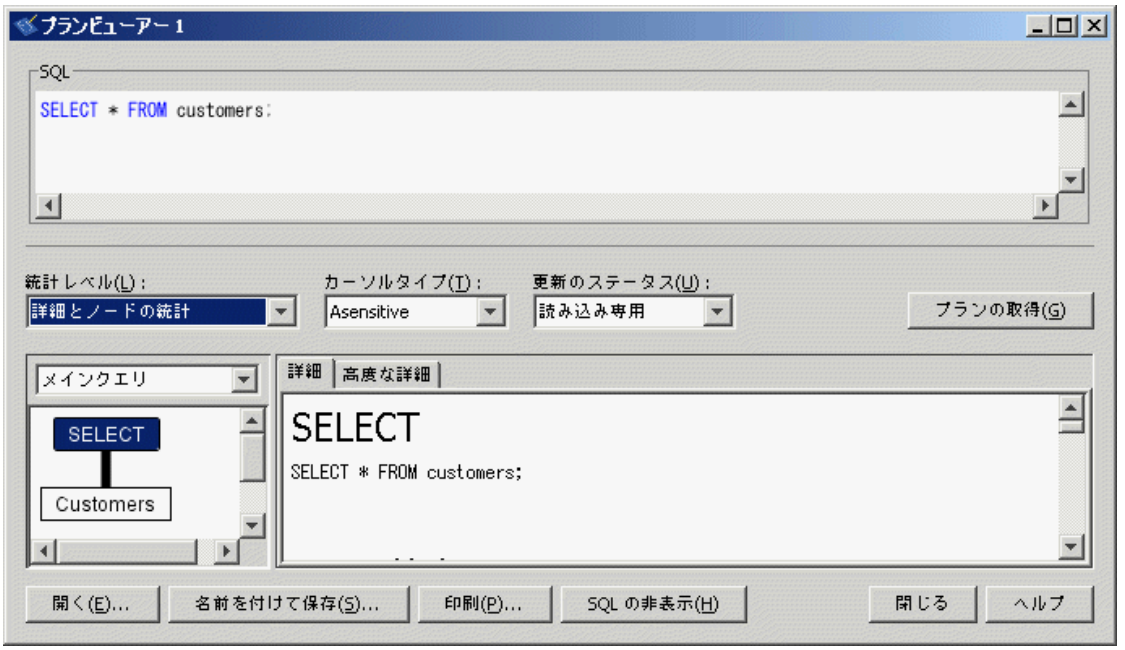
2. **[ツール]** » **[プランビューアーを開く]** をクリックするか、[Shift + F5] キーを押します。

   別ウィンドウにプランビューアーが表示されます。

3. **[SQL]** ウィンドウ枠に文を入力します。
4. **[統計レベル]** リストで、**[詳細とノードの統計]** をクリックします。
5. **[プランの取得]** をクリックして、指定したクエリのプランを生成します。
6. **[名前を付けて保存]** をクリックします。
7. プランの保存先を指定し、ファイル名を入力します。**[保存]** をクリックします。

プランビューアーのウィンドウは、次のウィンドウ枠に分かれています。

- **[SQL] ウィンドウ枠**　このウィンドウ枠には、プランを生成する SQL 文を入力します。
- **[結果] ウィンドウ枠**　このウィンドウ枠には、グラフィカルなプランが表示されます。これは SQL Anywhere データベース専用のウィンドウ枠です。
- **[詳細] ウィンドウ枠**　このウィンドウ枠には、SQL Anywhere データベースのプランに関する詳細テキストが表示されます。Ultra Light データベースでは、このウィンドウ枠にテキストプランが表示されます。




### ♦ グラフィカルなプランを開く

1. **[ツール]** » **[プランビューアーを開く]** をクリックします。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

2. **[開く]** をクリックします。
3. プランファイル (*.saplan*) を選択し、**[開く]** をクリックします。

**参照**

- 「グラフィカルなプラン」『SQL Anywhere サーバー SQL の使用法』

### グラフィカルなプランの設定

グラフィカルなプランの実行後に、プラン内の項目の表示をカスタマイズできます。

**♦ グラフィカルなプランの外観を変更する**

1. プランビューアーの左下のウィンドウ枠でプランを右クリックし、**[カスタマイズ]** をクリックします。
2. 設定を変更します。
3. 変更が完了したら **[OK]** をクリックします。
4. **[プランの取得]** をクリックして、変更を反映したグラフィカルなプランを生成します。

**参照**

- 「カスタマイズされたグラフィカルなプラン」『SQL Anywhere サーバー SQL の使用法』

## Interactive SQL の空間ビューアーとグラフィックイメージ

「空間データのイメージとしての表示 (Interactive SQL の場合)」『SQL Anywhere サーバー 空間データサポート』を参照してください。

## Interactive SQL とグラフィックイメージ

Interactive SQL を使用して、BLOB データと、5000000 ピクセル未満のイメージを表示できます。さらに、SVG を表示して、JPEG、PNG、TIFF などの他のフォーマットに保存できます。

サポートされているイメージフォーマットには、BMP (Windows 95 バージョン)、GIF、JPEG、PNG、PNM、SVG、TIFF、WBMP (ワイヤレスビットマップ) があります。

**♦ イメージと SVG を表示する (Interactive SQL の場合)**

1. データベースに接続します。たとえば、SQL Anywhere サンプルデータベースに接続します。
2. イメージが含まれる結果セットを返すクエリを実行します。

   たとえば、次のクエリを実行すると、JPEG イメージが返されます。

   ```
   SELECT * FROM Products;
   ```

たとえば、次のクエリを実行すると、SVG イメージが返されます。

```
SELECT '<?xml version="1.0" standalone="yes"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">

<rect width="99" height="99"
style="fill:rgb(0,0,255);stroke-width:1;
stroke:rgb(0,0,0)"/>

</svg>';
```

3. 結果セット内で、**(イメージ)** を含むセルをクリックし、**[...]** をクリックして **[ウィンドウで表示]** をクリックします。

   **[カラム *column-name* の値]** ウィンドウが表示されます。

   たとえば、コットン製のキャップに関連付けられた JPEG イメージをクリックすると、次のウィンドウが表示されます。

SVG を表示する場合は、次のショートカットを使用します。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| ショートカット | 説明 |
|---|---|
| [Ctrl] キーを押しながらマウスの左ボタンでドラッグ | 視点を調整できるように境界をドラッグします。 |
| [Ctrl] キーを押しながらマウスの右ボタンでドラッグ | 中心を軸にイメージを回転させます。 |
| [Shift] キーを押しながらマウスの左ボタンでドラッグ | イメージを新しいロケーションにパンします。 |
| [Shift] キーを押しながらマウスの右ボタンでドラッグ | イメージをズームイン、ズームアウトします。 |

4. **[OK]** をクリックして、ウィンドウを閉じます。

## Interactive SQL での HTML データと XML データの表示

**♦ HTML データと XML データを表示する (Interactive SQL の場合)**

1. データベースに接続します。

2. HTML データと XML データが含まれる結果セットを返すクエリを実行します。

   HTML が含まれるサンプルの結果セットを表示するには、次のクエリを実行します。

```
SELECT '<html>
 <head>
  <meta http-equiv=Content-Type content="text/html; charset=windows-1252">
   <title>Tee Shirt</title>
 </head>
 <body lang=EN-US>
  <p>
   <span  style=font-size:10.0pt;font-family:Arial>
   We have improved the design of this perennial favorite.
   A sleek and technical shirt built for the trail, track, or
   sidewalk. UPF rating of 50+.
   </span>
  </p>
 </body>
</html>';
```

   適切な形式の XML ドキュメントが含まれるサンプルの結果セットを表示するには、次のクエリを実行します。

```
SELECT '<?xml version="1.0" encoding="UTF-8"?>
<parent>
   <children>
      <child name="Tim" />
      <child name="Katie" />
   </children>
</parent>';
```

3. 結果内で、HTML コンテンツまたは XML コンテンツを含むセルをクリックし、**[...]** をクリックして **[ウィンドウで表示]** をクリックします。

   **[カラム *column-name* の値]** ウィンドウが表示されます。

4. **[OK]** をクリックして、ウィンドウを閉じます。

## Interactive SQL での SQL 文、実行プラン、結果セットの印刷

次の方法で、**[SQL 文]** ウィンドウ枠の内容またはクエリ結果を印刷できます。

●[Ctrl + P] キーを押す

●**[ファイル]** » **[印刷]** をクリックする

次の方法で、プランビューアーに表示されているプランを印刷できます。

●**[印刷]** をクリックする

●プランを右クリックし、**[印刷]** をクリックする

Interactive SQL の **[オプション]** ウィンドウでは、ヘッダーまたはフッターの追加や、その他のフォーマットオプションの設定もできます。

**♦ ヘッダーの追加**

1. Interactive SQL の **[オプション]** ウィンドウを開きます。

   **[ツール]** » **[オプション]** をクリックします。

2. **[エディター]** ページで **[印刷]** タブをクリックします。

3. **[ヘッダー]** フィールドで、ヘッダーに表示するテキストを指定します。右矢印をクリックして、ヘッダーに含める項目を選択することもできます。

## 複数の Interactive SQL ウィンドウを開く

複数の Interactive SQL ウィンドウを開くことができます。各ウィンドウは、接続するデータベース別になっています。異なるデータベースサーバー上にある 2 つ (またはそれ以上) のデータベースに同時に接続したり、単一のデータベースへの同時接続を開始したりできます。

**♦ 新しい Interactive SQL ウィンドウを開く**

1. **[ウィンドウ]** » **[新しいウィンドウ]** をクリックします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

> **ヒント**
> SQLCONNECT 環境変数が設定されている場合や、すでに SQL Anywhere データベースに接続している場合は、情報の入力を求める前に、データベースサーバーがこの情報を使ってデータベースに接続しようとします。同様に、ULSQLCONNECT 環境変数が設定されている場合や、すでに Ultra Light データベースに接続している場合は、情報の入力を求める前に、データベースサーバーがこの情報を使ってデータベースに接続しようとします。それに失敗した場合、またはデータベースにまだ接続していない場合に、**[接続]** ウィンドウが表示されます。

2. **[接続]** ウィンドウで、データベースの接続情報を入力し、**[接続]** をクリックします。

   接続情報 (データベース名、ユーザー ID、データベースサーバー名など) が Interactive SQL のタイトルバーに表示されます。

データベースとの接続または切断には、次のいずれかの方法も使用できます。

- **[SQL]** » **[接続]** または **[切断]** をクリックする
- CONNECT 文または DISCONNECT 文を実行する

**参照**

- 「SQLCONNECT 環境変数」455 ページ
- 「Ultra Light 接続パラメーターと ULSQLCONNECT 環境変数」『Ultra Light データベース管理とリファレンス』

## Interactive SQL でのソース制御の統合

Interactive SQL はサードパーティのソース制御システムと統合でき、Interactive SQL 内からファイルに対する一般的なソース制御操作を実行できます。Windows の場合、Interactive SQL は、Microsoft Visual SourceSafe などの Microsoft Common Source Code Control API (SCC) をサポートする、ほとんどのソース制御製品と統合できます。Windows やその他のオペレーティングシステムで SCC API をサポートしないソース制御製品を使用するには、ソース制御アクションごとに実行するコマンドラインを指定します。コマンドの出力は、ログウィンドウに表示されます。

Interactive SQL は、次のタスクをサポートします (対象のタスクがソース制御製品でサポートされている場合)。

- ソース制御プロジェクトを開く
- 取得
- チェックイン
- チェックアウト
- チェックアウトの取り消し
- バージョンの比較
- ファイルの履歴の表示
- ファイルのプロパティの表示
- ソース制御マネージャーの実行

基本となるソース制御プログラムがアクションをサポートしていない場合は、対応するメニュー項目が無効になります。たとえば、Microsoft Visual SourceSafe は上記のすべてのアクションをサ

ポートしていますが、カスタム (コマンドライン) ソース制御システムを使用すると、ソース制御プロジェクトを開いたり、ソース制御マネージャーを実行したりすることはサポートされません。

Interactive SQL からソース制御プログラムを使用する前に、その操作を理解しておくことが必要です。

## Interactive SQL でのソース制御

ファイルのチェックインやチェックアウト、異なるバージョンのファイルの比較、ファイルの履歴の表示など、ファイルに対するソース制御アクションを実行するには、事前に、ソース制御を使用するように Interactive SQL を設定する必要があります。

Microsoft SCC API をサポートするソース制御製品を備えた Windows コンピューター上で Interactive SQL を実行している場合は、その製品またはカスタム (コマンドライン指向の) システムを使用できます。

### SCC ソース制御システムの設定

**♦ SCC を備えた Windows での Interactive SQL ソース制御を設定する**

1. **[ツール]** » **[オプション]** をクリックします。
2. 左ウィンドウ枠で、**[ソース制御]** をクリックします。
3. **[ソース制御の統合を有効にする]** をクリックします。
4. **[OK]** をクリックします。

### その他のソース制御システムの設定

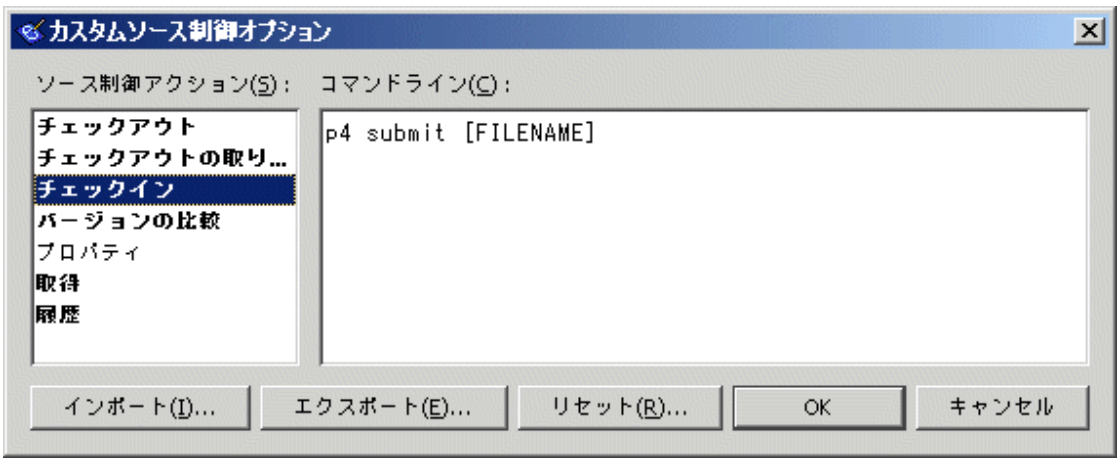
**♦ コマンドラインインターフェイスを備えた Interactive SQL ソース制御システムを設定する**

1. **[ツール]** » **[オプション]** をクリックします。
2. 左ウィンドウ枠で、**[ソース制御]** をクリックします。
3. **[ソース制御の統合を有効にする]** をクリックします。
4. **[設定]** をクリックします。
5. **[カスタムソース制御オプション]** ウィンドウで **[リセット]** をクリックします。
6. リストからソース制御システムを選択し、**[OK]** をクリックします。
7. 必要に応じてリスト内のコマンドを編集します。これを行うには、**[ソース制御アクション]** リストからアクションを選択し、**[コマンドライン]** ウィンドウ枠に対応するコマンドを入力します。

   **[ソース制御アクション]** リストで使用システム用のコマンドを定義する場合は、プレースホルダー [FILENAME] を使用して、そのコマンドの実行時に使用するファイル名を表します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

たとえば、Perforce でファイルを送信するためのコマンドは p4 submit [FILENAME] です。リスト内で太字で示されているアクションには、対応するコマンドが定義されています。標準のフォントで示されているアクションには、対応するコマンドが定義されていません。




アクションに対応するコマンドラインを指定しないと、**[ファイル]** » **[ソース制御]** メニューの項目は無効になります。

> **ヒント**
> **[カスタムソース制御オプション]** ウィンドウの **[エクスポート]** をクリックすると、ソース制御コマンドラインを外部ファイルにエクスポートできます。このウィンドウにアクセスするには、**[ツール]** » **[オプション]** をクリックし、**[ソース制御]** ウィンドウ枠で **[設定]** をクリックします。[カスタムソース制御オプション] ウィンドウで **[インポート]** をクリックすると、後でこれらのコマンドを読み込むことができます。この機能は、複数のコンピューター上で Interactive SQL ソース制御コマンドラインを設定する必要がある場合に便利です。

8. **[OK]** をクリックし、もう一度 **[OK]** をクリックします。

## ソース制御プロジェクトと Interactive SQL

ソース制御製品の中には、ソース制御アクションを実行する前にソース制御プロジェクトを開くことが必要なものもあります。プロジェクトの厳密な定義は、使用しているソース制御システムによって異なります。一般には、ソース制御下に置かれているファイルのセットと、それらファイルの作業用コピーが配置されているローカルファイルシステム上のロケーションのことです。プロジェクトを開くには、通常、ユーザー ID とパスワードのようなクレデンシャルをソース制御システムに提供する必要があります。

使用しているソース制御システムでソース制御プロジェクトを開くことがサポートされている場合は、**[ファイル]** » **[ソース制御]** » **[ソース制御プロジェクトを開く]** メニュー項目が有効になります。**[ファイル]** メニューからこのオプションを選択すると、プロジェクトを開くための、ソース制御に固有のウィンドウが開きます。一度プロジェクトを開くと、以降の Interactive SQL セッションであっても、同じプロジェクトを再び開く必要はありません。プロジェクトは自動的に開きます。

## Interactive SQL の使用によるファイルのチェックアウト

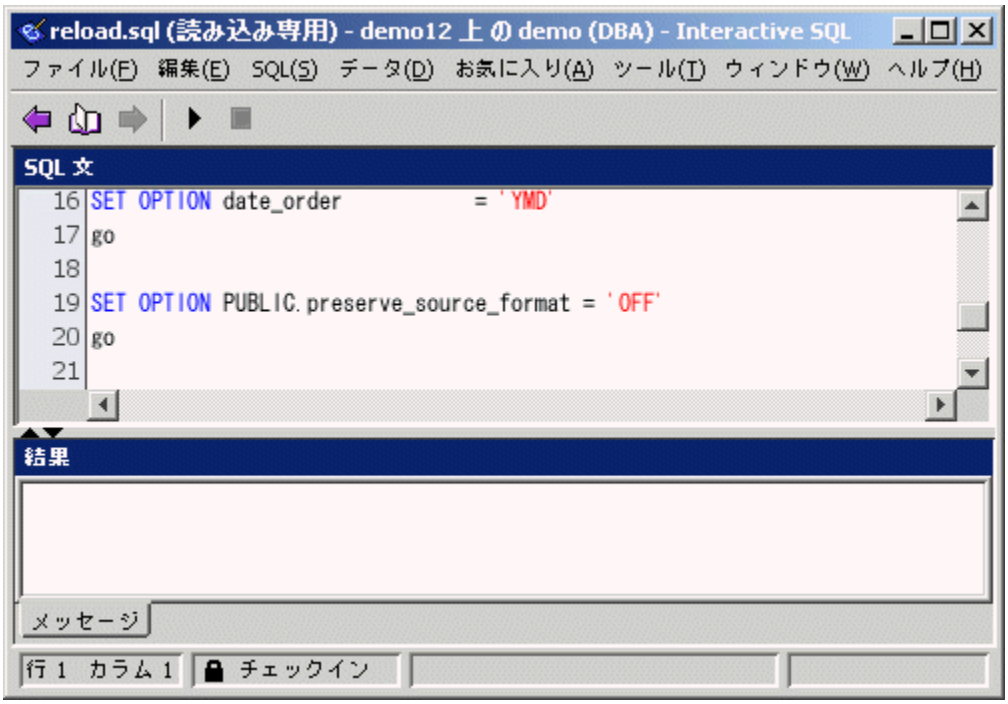
Interactive SQL でファイルを開いた後に、ファイルをチェックアウトするには 2 つの方法があります。**[SQL 文]** ウィンドウ枠で内容を変更する方法と、**[ファイル]** メニューのコマンドを使用する方法です。

**[SQL 文]** ウィンドウ枠でファイルの内容を変更したときに、Interactive SQL によってそのファイルがチェックアウトされるようにするには、Interactive SQL のソース制御オプションを構成するときに **[エディターの内容が変更されたら自動的にファイルをチェックアウト]** を選択します。

### ♦ Interactive SQL の [ファイル] メニューを使用してファイルをチェックアウトする

1. **[ファイル]** » **[開く]** をクリックし、開きたいファイルを参照します。

   Interactive SQL ウィンドウ下部にあるステータスバーに、ファイルのステータスが表示されます。ステータスは、**[チェックイン]**、**[チェックアウト]**、**[Not Controlled]** のいずれかになります。チェックインされているファイルは読み込み専用と見なされ、Interactive SQL のタイトルバーに **[読み込み専用]** と表示されます。次の例に、チェックインされているファイルを示します。




2. **[ファイル]** » **[ソース制御]** » **[チェックアウト]** をクリックして、ファイルをチェックアウトします。

   使用しているソース制御製品によっては、チェックアウト手順でコメントの入力やその他のオプション操作を求められることがあります。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**警告**
SCC に準拠するソース制御システムを使用している場合、必ず正しいステータスが表示されます。ただし、カスタムソース制御システムを使用している場合は、ファイルが読み込み専用であるかどうかによってステータスが表示されます。読み込み専用のファイルはチェックインされていると見なされますが、編集可能なファイルについてはこの前提条件は適用されません。これは、編集可能なファイルはチェックアウト可能であるか、制御不能であるためです。

## Interactive SQL の使用によるファイルのチェックイン

ファイルの編集が完了したら、Interactive SQL からチェックインできます。

**♦ Interactive SQL からファイルをチェックインする**

1. **[ファイル]** » **[ソース制御]** » **[チェックイン]** をクリックします。
2. プロンプトが表示された場合は、チェックインについてのコメントを入力します。

## その他のソース制御アクション

ソース制御プロジェクトを開いたり、ファイルのチェックインやチェックアウトを行ったりすることに加えて、Interactive SQL はその他のソース制御アクションをサポートします。これらのアクションの可用性は、使用しているソース制御システムによって異なります。**[ファイル]** » **[ソース制御]** をクリックしてから、次のいずれかのアクションを選択します。

- **[取得]** このアクションは、**[SQL 文]** ウィンドウ枠で現在開いているファイルの最新のコピーを取得します。
- **[チェックアウトの取り消し]** ファイルをチェックアウトしてから変更を破棄するには、**[ファイル]** » **[ソース制御]** » **[チェックアウトの取り消し]** をクリックします。この操作によって、ファイルの作業用コピーは破棄され、ソース制御のアーカイブにあるファイルのコピーがダウンロードされます。
- **[バージョンの比較]** このアクションは、開いているファイルの作業用コピーと、ソース制御のアーカイブにあるファイルのバージョンを比較します。
- **[履歴]** このアクションは、開いているファイルに対して実行されたソース制御アクション (通常はチェックイン) のリストを表示します。
- **[プロパティ]** このアクションは、開いているファイルに関連のあるソース制御プロパティのリストを表示します。
- **[ソース制御マネージャーの実行]** このアクションは、使用しているソース制御システムの管理プログラムを起動します。たとえば、Microsoft Visual SourceSafe を使用している場合、Microsoft Visual SourceSafe エクスプローラーを起動します。

## Interactive SQL の SQL 文

次のリストは、Interactive SQL ユーザーが使用できる SQL 文について説明している SQL Anywhere マニュアルへのリンクです。

- 「CLEAR 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「CONFIGURE 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「CONNECT 文 [ESQL] [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「DESCRIBE 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「DISCONNECT 文 [ESQL] [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「EXIT 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「HELP 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「INPUT 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「OUTPUT 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「PARAMETERS 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「READ 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「SET CONNECTION statement [Interactive SQL] [ESQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「START SERVER 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「START LOGGING 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「STOP LOGGING 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「SYSTEM 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』

**参照**

- 「SQL 文」『SQL Anywhere サーバー SQL リファレンス』
- 「Ultra Light SQL 文」『Ultra Light データベース管理とリファレンス』

## Interactive SQL のキーボードショートカット

Interactive SQL には、以下のキーボードショートカットがあります。

| キー | 説明 |
|---|---|
| [Alt + F4] | Interactive SQL を停止します。 |
| [Alt + 左矢印]<br><br>Mac OS X では、[Control-Command + 左矢印] を使用します。 | 履歴リストにある、前の SQL 文を表示します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| キー | 説明 |
|---|---|
| [Alt + 右矢印]<br><br>Mac OS X では、[Control-Command + 右矢印] を使用します。 | 履歴リストにある、次の SQL 文を表示します。 |
| [Ctrl + Backspace] | カーソルの左の 1 ワードを削除します。 |
| [Ctrl + Break] | 実行中の SQL 文を中断します。 |
| [Ctrl + A] | アクティブなウィンドウ枠のすべてのテキストを選択します。<br><br>**[結果]** ウィンドウ枠に表示されている結果が結果セット全体でない場合、残りのローをフェッチするよう要求されます。残りをフェッチしないと、現在フェッチされている結果が選択されます。「Interactive SQL の結果セットからカラム、ロー、セルをコピーする」862 ページを参照してください。 |
| [Ctrl + C] | **[結果]** ウィンドウ枠では、選択したローとカラム見出しをクリップボードにコピーします。<br><br>**[SQL 文]** ウィンドウ枠では、選択したテキストをクリップボードにコピーします。 |
| [Ctrl + Shift + C] | COMMIT 文を実行します。「Interactive SQL での COMMIT 文と ROLLBACK 文の実行」851 ページを参照してください。 |
| [Ctrl + Del] | カーソルの右の 1 ワードを削除します。 |
| [Ctrl + End] | カーソルを現在のウィンドウ枠の一番下に移動します。 |
| [Ctrl + F] | **[検索／置換]** ウィンドウを開きます。 |
| [Ctrl + Shift + F10] | **[結果]** タブの結果セット内にフォーカスがあるセルのポップアップメニューを表示します。<br><br>このキーボードショートカットは、テーブルセル内の **[...]** をクリックする代わりに使用します。 |
| [Ctrl + G] | カーソルを **[SQL 文]** ウィンドウ枠内の指定の行に移動します。 |
| [Ctrl + H]<br><br>Mac OS X では、[Control-Command + H] を使用します。 | 実行した SQL 文の履歴を表示します。 |

| キー | 説明 |
|---|---|
| [Ctrl + Home] | カーソルを現在のウィンドウ枠の先頭に移動します。 |
| [Ctrl + L] | 現在の行を **[SQL 文]** ウィンドウ枠から削除してクリップボードに入れます。 |
| [Ctrl + Shift + L] | 現在の行を削除します。 |
| [Ctrl + N] | Interactive SQL ウィンドウの内容をクリアし、現在のファイルを閉じます (存在する場合)。 |
| [Ctrl + O] | ファイルを開きます。 |
| [Ctrl + P] | **[SQL 文]** ウィンドウ枠の内容を印刷します。または、**[ツール]** » **[オプション]** » **[エディター]** » **[印刷]** をクリックします。 |
| [Ctrl + Q] | クエリエディターを開きます。<br><br>クエリエディターを使用すると、SQL クエリを作成できます。クエリの構築が完了したら、**[OK]** をクリックして、クエリを **[SQL 文]** ウィンドウ枠にエクスポートします。 |
| [Ctrl + Shift + R] | ROLLBACK 文を実行します。「Interactive SQL での COMMIT 文と ROLLBACK 文の実行」851 ページを参照してください。 |
| [Ctrl + S] | **[SQL 文]** ウィンドウ枠の内容を指定ファイルに保存します。 |
| [Ctrl + U] | 選択している内容を小文字に変換します。 |
| [Ctrl + Shift + U] | 選択している内容を大文字に変換します。 |
| [Ctrl + V] | 選択したテキストを貼り付けます。 |
| [Ctrl + X] | 選択したテキストを切り取ります。 |
| [Ctrl + Y] | 最後の操作を繰り返します。 |
| [Ctrl + Z] | 最後の操作を取り消します。 |
| [Ctrl + ]] | カーソルを閉じカッコまで移動します。カッコ、大カッコ、角カッコ、山カッコに一致させるときに、このショートカットを使用します。 |
| [Ctrl + Shift + ]] | 選択範囲を閉じカッコまで拡張します。カッコ、大カッコ、角カッコ、山カッコに一致させるときに、このショートカットを使用します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| キー | 説明 |
|---|---|
| [Ctrl + マイナス記号 (-)] | SQL コメントインジケーターの二重ハイフン (--) を追加および削除します。<br><br>既存のテキストをコメントに変換するには、**[SQL 文]** ウィンドウ枠でテキストを選択し、[Ctrl + マイナス記号] キーを押します。SQL コメントインジケーターが、選択したテキストの各行の先頭に追加されます。<br><br>テキストが選択されていない場合、コメントインジケーターは現在の行の先頭に追加されます。<br><br>コメントインジケーターを削除するには、テキストを選択して [Ctrl + マイナス記号] キーを押します。<br><br>「コメント」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| [Ctrl + スラッシュ (/)] | SQL コメントインジケーターの二重スラッシュ (//) を追加および削除します。<br><br>既存のテキストをコメントに変換するには、**[SQL 文]** ウィンドウ枠でテキストを選択し、[Ctrl + スラッシュ] キーを押します。SQL コメントインジケーターが、選択したテキストの各行の先頭に追加されます。<br><br>テキストが選択されていない場合、コメントインジケーターは現在の行の先頭に追加されます。<br><br>コメントインジケーターを削除するには、テキストを選択して [Ctrl + スラッシュ] キーを押します。<br><br>「コメント」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| [Ctrl + 上矢印] | **[SQL 文]** ウィンドウ枠で、カーソルがある文の前の SQL 文を選択します。 |
| [Ctrl + 下矢印] | **[SQL 文]** ウィンドウ枠で、カーソルがある文の後の SQL 文を選択します。 |
| [Ctrl + ピリオド (.)] | **[SQL 文]** ウィンドウ枠で、カーソルがある SQL 文全体を選択します。 |
| [Ctrl + カンマ (,)] | **[SQL 文]** ウィンドウ枠で、カーソルがある行を選択します。 |

| キー | 説明 |
| --- | --- |
| [Ctrl + Shift + ピリオド(.)] | **[SQL 文]** ウィンドウ枠で選択されたテキストの行インデントを増加します。<br><br>テキストが選択されていない場合、インデントは現在の行に適用されます。<br><br>インデントするスペースの数を変更できます。または、**[ツール]** » **[オプション]** » **[エディター]** » **[タブ]** をクリックします。 |
| [Ctrl + Shift + カンマ(,)] | **[SQL 文]** ウィンドウ枠で選択されたテキストの行インデントを減少します。<br><br>テキストが選択されていない場合、インデントは現在の行に適用されます。<br><br>インデントするスペースの数を変更できます。または、**[ツール]** » **[オプション]** » **[エディター]** » **[タブ]** をクリックします。 |
| [Esc] | デフォルトでは、[Esc] キーを押しても、何も起こりません。<br><br>ただし、**[SQL 文]** ウィンドウ枠の内容をクリアし、開いている結果セットを閉じるように [Esc] キーを設定できます。**[ツール]** » **[オプション]** » **[互換性]** をクリックし、**[[Esc] キーで SQL 文をクリアし、結果セットを閉じる]** をクリックします。 |
| F1 | オンラインヘルプを表示します。 |
| [F2] | 結果セット内の選択した値を編集します。ローのカラム間でタブ移動できます。 |
| F3 | 指定したテキストの次の出現箇所を検索します。 |
| [Shift + F3] | 選択したテキストの 1 つ前の出現箇所を検索します。 |
| F5 | **[SQL 文]** ウィンドウ枠にあるすべてのテキストを実行します。<br><br>この操作は、ツールバーで **[すべての SQL 文の実行]** をクリックするか、**[SQL]** » **[実行]** をクリックしても実行できます。 |
| [Shift + F5] | **[SQL 文]** ウィンドウ枠で指定された文のプランビューアーを開きます。指定された文は実行されません。その文をプランビューアーで実行するには、**[プランの取得]** をクリックします。 |
| [F6] | フォーカスを **[SQL 文]** ウィンドウ枠から **[結果]** ウィンドウ枠に、またはその逆に移動します。**[お気に入り]** ウィンドウ枠が開いている場合、フォーカスは **[SQL 文]**、**[結果]**、**[お気に入り]** の各ウィンドウ枠間を移動します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| キー | 説明 |
|---|---|
| [F7] | **[テーブル名のルックアップ]** ウィンドウを表示します。<br><br>このウィンドウでテーブルを検索して選択し、[Enter] キーを押せば、そのテーブル名が **[SQL 文]** ウィンドウ枠のカーソル位置に入力されます。また、リストでテーブルを選択し、もう一度 [F7] を押すとそのテーブルのカラムが表示されます。この後カラムを選択して [Enter] キーを押せば、そのカラム名が **[SQL 文]** ウィンドウ枠のカーソル位置に入力されます。 |
| [F8] | **[プロシージャー名のルックアップ]** ウィンドウを表示します。<br><br>このウィンドウでプロシージャーを検索して選択し、[Enter] キーを押せば、そのプロシージャー名が **[SQL 文]** ウィンドウ枠のカーソル位置に入力されます。 |
| F9 | **[SQL 文]** ウィンドウ枠で選択されているテキストを実行します。<br><br>テキストが選択されていない場合は、すべての文が実行されます。<br><br>この操作は、ツールバーで **[選択した文の実行]** をクリックするか、**[SQL]** » **[実行]** をクリックしても実行できます。 |
| [Shift + F9] | 選択された SQL 文を実行し、次の文を選択します。このショートカットを使用すると、一連の SQL 文をステップスルーできます。<br>「Interactive SQL での SQL 文の実行」848 ページを参照してください。 |
| [Shift + F10] | フォーカスのある領域のショートカットメニューを表示します。<br><br>このキーボードショートカットは、領域を右クリックする代わりに使用します。 |
| F11 | Interactive SQL がデータベースに接続されていない場合に **[接続]** ウィンドウを開きます。 |
| F12 | Interactive SQL と現在のデータベースの接続を切断します。 |
| [Home] | カーソルを現在の行の行頭、または現在の行の最初の単語に移動します。 |
| [Shift + Home] | 選択範囲を現在の行のテキストの先頭まで拡張します。 |
| [Page Down] | 現在のウィンドウ枠を 1 ページ下にスクロールします。 |
| [Page Up] | 現在のウィンドウ枠を 1 ページ上にスクロールします。 |

**参照**

- 「テキスト補完キーボードショートカット」902 ページ

# Interactive SQL オプション

SET OPTION 文は、Interactive SQL の次のオプションの値を変更するために使用します。「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

| オプション | 値 | デフォルト |
|---|---|---|
| 「auto_commit オプション [Interactive SQL]」 | On、Off | Off |
| 「auto_refetch オプション [Interactive SQL]」 | On、Off | On |
| 「bell オプション [Interactive SQL]」 | On、Off | On |
| 「command_delimiter オプション [Interactive SQL]」 | 文字列 | ';' |
| 「commit_on_exit オプション [Interactive SQL]」 | On、Off | On |
| 「default_isql_encoding オプション [Interactive SQL]」 | 文字列 | 空の文字列 |
| 「echo オプション [Interactive SQL]」 | On、Off | On |
| 「input_format オプション [Interactive SQL]」 | TEXT、FIXED | TEXT |
| 「isql_allow_read_client_file オプション [Interactive SQL]」 | On、Off、Prompt | Prompt |
| 「isql_allow_write_client_file オプション [Interactive SQL]」 | On、Off、Prompt | Prompt |
| 「isql_command_timing オプション [Interactive SQL]」 | On、Off | On |
| 「isql_escape_character オプション [Interactive SQL]」 | 文字 | '¥' |
| 「isql_field_separator オプション [Interactive SQL]」 | 文字列 | ',' |
| 「isql_maximum_displayed_rows オプション [Interactive SQL]」 | All または負でない整数 | 500 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 値 | デフォルト |
|---|---|---|
| 「isql_print_result_set オプション [Interactive SQL]」 | Last、All、None | Last |
| 「isql_quote オプション [Interactive SQL]」 | 文字列 | ' |
| 「isql_show_multiple_result_sets オプション [Interactive SQL]」 | On、Off | Off |
| 「nulls オプション [Interactive SQL]」 | 文字列 | (NULL)' |
| 「on_error オプション [Interactive SQL]」 | Stop、Continue、Prompt、Exit、Notify_Continue、Notify_Stop、Notify_Exit | Prompt |
| 「output_format オプション [Interactive SQL]」 | TEXT、FIXED、HTML、SQL、XML | TEXT |
| 「output_length オプション [Interactive SQL]」 | 整数 | 0 |
| 「output_nulls オプション [Interactive SQL]」 | 文字列 | 空の文字列 |
| 「truncation_length オプション [Interactive SQL]」 | 整数 | 256 |

## auto_commit オプション [Interactive SQL]

各文の後で COMMIT が実行されるかどうかを制御します。

### 指定可能な値

On、Off

### デフォルト

Off

### 備考

auto_commit が On の場合、各文の処理が成功した後でデータベースの COMMIT が実行されます。

デフォルトでは、COMMIT または ROLLBACK が実行されるのは、ユーザーが COMMIT 文または ROLLBACK 文を発行するか、自動コミットを行う SQL 文 (CREATE TABLE 文など) を発行したときだけです。

> **注意**
> デフォルトでは、ODBC はオートコミットモードで動作します。Interactive SQL で auto_commit オプションを Off に設定しても、ODBC 設定が Interactive SQL の設定を上書きします。ODBC の設定は、SQL_ATTR_AUTOCOMMIT 接続属性を使用して変更できます。ODBC オートコミットは chained オプションから独立しています。

### 参照

- 「commit_on_exit オプション [Interactive SQL]」888 ページ
- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「Interactive SQL での COMMIT 文と ROLLBACK 文の実行」851 ページ

## auto_refetch オプション [Interactive SQL]

削除、更新、または挿入後にクエリ結果が再度フェッチされるかどうかを制御します。

### 指定可能な値

On、Off

### デフォルト

On

### 戻り値

auto_refetch が On の場合、Interactive SQL の **[結果]** ウィンドウ枠の **[結果]** タブに表示される現在のクエリ結果は、INSERT 文、UPDATE 文、または DELETE 文の後でデータベースから再フェッチされます。クエリの複雑さによって、時間のかかるものもあります。このため、これをオフにすることもできます。

### 参照

- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』

## bell オプション [Interactive SQL]

エラーが起こったときにベルを鳴らすかどうかを制御します。

### 指定可能な値

On、Off

### デフォルト

On

### 戻り値

このオプションは好みに応じて設定します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照

- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』

## command_delimiter オプション [Interactive SQL]

Interactive SQL で文の終わりを示す文字列を設定します。

### 指定可能な値

文字列

### デフォルト

セミコロン (;)

### 戻り値

通常は、コマンドデリミターを変更する必要はありません。セミコロンのままにしてください。

文のデリミターとしてセミコロンなどの文字列を使用する代わりに、セパレーター **go** を独立した行の先頭に配置することもできます。「バッチ」『SQL Anywhere サーバー SQL の使用法』を参照してください。

独立した行の先頭に指定された **go** は、command_delimiter オプションの値にかかわらず常にコマンドデリミターとして認識されます。

command_delimiter オプションには任意の文字列を指定できます。ただし、次のような制限があります。

- & (アンパサンド)、* (アスタリスク)、@ (アットマーク)、: (コロン)、 (ピリオド)、= (等号)、( (左カッコ)、) (右カッコ)、または | (パイプ記号) が含まれているデリミターには、それ以外の文字を追加することはできません。たとえば、* は有効なデリミターですが、** はデリミターとして無効です。

- 既存のキーワードをコマンドセパレーターとして使用しないでください。

- コマンドデリミターには、数字、文字、句読点などの任意の文字列を使用できますが、埋め込みブランクを含めることはできません。また、セミコロンは、先頭文字としてだけ含めることができます。

  コマンドデリミターとして設定されている文字列が、識別子として有効な文字で始まる場合は、前にスペースを付けてください。コマンドデリミターでは、大文字と小文字が区別されます。新しいコマンドデリミターは一重引用符で囲んでください。コマンドデリミターがセミコロン (デフォルト) の場合、セミコロンの前にスペースを入れる必要はありません。

### 参照

- 「キーワード」『SQL Anywhere サーバー SQL リファレンス』
- 「Interactive SQL ユーティリティ (dbisql)」975 ページ
- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』

### 例

次の例は、コマンドデリミターをチルダに設定します。

```
SET OPTION command_delimiter='~';
MESSAGE 'hello'~
```

Interactive SQL の -d オプションを使用してコマンドデリミターを設定することもできます。この場合、*.sql* ファイルに SET OPTION command_delimiter 文を記述する必要はありません。たとえば、スクリプトファイル *test.sql* で、チルダ (~) をコマンドデリミターとして使用する場合は、次のようになります。

```
dbisql -d "~" test.sql
```

## commit_on_exit オプション [Interactive SQL]

Interactive SQL が切断または停止されるときの動作を制御します。

### 指定可能な値

On、Off

### デフォルト

On

### 戻り値

Interactive SQL を終了するときに COMMIT か ROLLBACK を行うかを制御します。commit_on_exit を On に設定すると、COMMIT が行われます。

### 参照

- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「Interactive SQL での COMMIT 文と ROLLBACK 文の実行」851 ページ

## default_isql_encoding オプション [Interactive SQL]

READ、INPUT、OUTPUT 文で使用されるエンコードを指定します。

### 指定可能な値

識別子または文字列

### デフォルト

システムエンコード (空の文字列) を使用します。

### スコープ

現在の接続の間、テンポラリオプションとしてのみ設定できます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 戻り値

このオプションは、ファイルを読み込みまたは書き込みするときに使用するエンコードを指定するのに使用されます。これを永続的に設定することはできません。デフォルトのエンコードは、実行しているプラットフォームのコードページです。英語版 Windows コンピューターでは、デフォルトのコードページは 1252 です。

Interactive SQL を実行している場合、INPUT 文、OUTPUT 文、または READ 文で使用されるエンコードは、次の順序で決定されます。

- INPUT 文、OUTPUT 文、または READ 文の ENCODING 句に指定されたエンコード (この句が指定されている場合)。
- default_isql_encoding オプションで指定されたエンコード (このオプションが設定されている場合)。
- 実行しているプラットフォームのデフォルトのエンコード。英語版 Windows コンピューターでは、デフォルトのエンコードは 1252 です。

OUTPUT 文を使用して入力ファイルを作成したときにエンコードを指定した場合は、INPUT 文でも同じ ENCODING 句を指定してください。

コードページと文字セットの詳細については、「国際言語と文字セットのタスク」501 ページを参照してください。

### 参照

- 「OUTPUT 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「READ 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「INPUT 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「文字セット」473 ページ
- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』

### 例

コード化を UTF-16 (Unicode ファイルの読み込み用) に設定します。

```
SET TEMPORARY OPTION default_isql_encoding = 'UTF-16';
```

エンコードを cp437 (英語版 Windows OEM) に設定します。

```
SET TEMPORARY OPTION default_isql_encoding = 'cp437';
```

## echo オプション [Interactive SQL]

文を実行する前にそれをログファイルへエコーするかどうかを制御します。

### 指定可能な値

On、Off

### デフォルト

On

### 戻り値

このオプションは、READ 文を使用して SQL スクリプトファイルを実行する場合、または Interactive SQL で **[ファイル]** » **[スクリプトの実行]** をクリックしてスクリプトファイルを実行する場合に最適です。このオプションを有効にするには、ロギングをオンにする必要があります。「START LOGGING 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

### 参照

- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』

## input_format オプション [Interactive SQL]

INPUT 文が予期するデフォルトデータフォーマットを設定します。

### 指定可能な値

- **TEXT**　入力行はテキスト文字であり、1 行あたり 1 つのローで構成され、値はカンマで区切られているものと見なされます。アルファベットの文字列をアポストロフィ (一重引用符) または二重引用符で囲むことができます。

  デフォルトのデリミターはカンマ (,) です。カンマを含む文字列は、一重引用符または二重引用符のどちらかで囲む必要があります。一重か二重引用符を使用している場合、文字列内で使用するには引用符を 2 つ重ねてください。オプションで、DELIMITED BY 句を使用して、別のデリミターを指定できます。

  他の 3 つの特別なシーケンスも認識できます。2 つの文字 **¥n** は改行文字を表し、**¥¥** は 1 つの円記号 (¥) を表し、シーケンス **¥x***DD* (*DD* は文字の 16 進表現) は 16 進コード DD の文字を表します。

- **FIXED**　入力行は固定長フォーマットです。

### デフォルト

TEXT

### 参照

- 「INPUT 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』

## isql_allow_read_client_file オプション [Interactive SQL]

接続でのクライアントファイルの読み込みを許可するかどうかを制御します。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 指定可能な値

On、Off、Prompt

### デフォルト

Prompt

### 戻り値

このオプションは、データベースサーバーがクライアントコンピューター上のファイルを読み込めるかどうかを制御します。On に設定すると、読み込みが許可されます。Off に設定すると、読み込みは許可されません。Prompt に設定すると、プロンプトが表示され、ユーザーが動作を指定できます。

このオプションは接続単位で保存され、その接続の間だけ持続します。このオプションは、SET TEMPORARY OPTION 文を使用して設定します。TEMPORARY キーワードを省略すると、Interactive SQL からエラーがレポートされます。

このオプションを使用すると、ストアドプロシージャーまたはトリガーから LOAD TABLE が実行された場合に、ユーザーの介入なしにデータファイルを読み込めます。

クライアントコンピューターにあるファイルを読み込むには、READCLIENTFILE 権限が必要です。

### 参照

- 「クライアントコンピューター上のデータへのアクセス」『SQL Anywhere サーバー SQL の使用法』
- 「READCLIENTFILE 権限」540 ページ
- 「READ_CLIENT_FILE 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』
- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「LOAD TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「allow_read_client_file オプション」590 ページ
- 「allow_write_client_file オプション」592 ページ
- 「isql_allow_write_client_file オプション [Interactive SQL]」891 ページ
- 「クライアント側データセキュリティ」『SQL Anywhere サーバー SQL の使用法』

## isql_allow_write_client_file オプション [Interactive SQL]

接続でのクライアントファイルの書き込みを許可するかどうかを制御します。

### 指定可能な値

On、Off、Prompt

### デフォルト

Prompt

### 戻り値

このオプションは、データベースサーバーからクライアントコンピューター上のファイルに書き込めるかどうかを制御します。On に設定すると、書き込みが許可されます。Off に設定すると、書き込みは許可されません。Prompt に設定すると、プロンプトが表示され、ユーザーが動作を指定できます。

このオプションは接続単位で保存され、その接続の間だけ持続します。このオプションは、SET TEMPORARY OPTION 文を使用して設定します。TEMPORARY キーワードを省略すると、Interactive SQL からエラーがレポートされます。

クライアントコンピューターにファイルを書き込むには、WRITECLIENTFILE 権限が必要です。

### 参照

- 「クライアントコンピューター上のデータへのアクセス」『SQL Anywhere サーバー SQL の使用法』
- 「WRITECLIENTFILE 権限」541 ページ
- 「WRITE_CLIENT_FILE 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』
- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「UNLOAD 文」『SQL Anywhere サーバー SQL リファレンス』
- 「allow_write_client_file オプション」592 ページ
- 「allow_read_client_file オプション」590 ページ
- 「isql_allow_read_client_file オプション [Interactive SQL]」890 ページ
- 「クライアント側データセキュリティ」『SQL Anywhere サーバー SQL の使用法』

## isql_command_timing オプション [Interactive SQL]

SQL 文の実行時間を記録するかどうかを制御します。

### 指定可能な値

On、Off

### デフォルト

On

### 戻り値

このブールオプションでは、SQL 文の実行時間を記録するかどうかを制御します。このオプションを On に設定すると、SQL 文を実行した後に、実行時間が **[メッセージ]** ウィンドウ枠に表示されます。Off に設定すると、実行時間は表示されません。

ロギングと実行時間の両方のレポートが有効になっている場合は、ログファイルに実行時間が記録されます。「START LOGGING 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

**[メッセージ]** タブでは、isql_command_timing オプションは **[SQL 文の実行時間を計測]** オプションとして表示されます。**[ツール]** » **[オプション]** » **[メッセージ]** をクリックし、**[SQL 文の実行時間を計測]** オプションを選択するか、選択を解除します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 参照

- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』

## isql_escape_character オプション [Interactive SQL]

テキストファイルにエクスポートされたデータに印刷不能な文字が含まれていた場合に、その代わりに使用するエスケープ文字を制御します。

### 指定可能な値

任意の 1 文字

### デフォルト

円記号 (¥)

### 戻り値

Interactive SQL が改行など印刷不能な文字を含む文字列をエクスポートする場合、印刷不能な文字は 16 進数フォーマットに変換され、その文字の前にはエスケープ文字が付加されます。OUTPUT 文に ESCAPE CHARACTER 句がない場合に、この設定で指定した文字が出力で使用されます。この設定は、テキストファイルにエクスポートする場合にだけ使用されます。

### 参照

- 「isql_quote オプション [Interactive SQL]」896 ページ
- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』

### 例

改行が埋め込まれた 1 つの文字列値を含む表を作成します (INSERT 文に "¥n" で示されます)。次に、# 記号をエスケープ文字として使用して、データを *c:¥escape.txt* にエクスポートします。

```
CREATE TABLE escape_test( text varchar(10 ) );
INSERT INTO escape_test VALUES( 'one¥ntwo' );
SET OPTION isql_escape_character='#';
SELECT * FROM escape_test;
OUTPUT TO c:¥escape.txt FORMAT TEXT;
```

このコードを実行すると、次のデータが *escape.txt* に書き込まれます。

```
'one#x0atwo'
```

シャープ記号 (#) はエスケープ文字であり、**x0a** は **¥n** の 16 進表現です。

先頭と末尾の文字 (この場合は一重引用符) は、isql_quote の設定によって異なります。

## isql_field_separator オプション [Interactive SQL]

テキストファイルにエクスポートするデータの値を区切るのに使用される、デフォルトの文字列を制御します。

### 指定可能な値

文字列

### デフォルト

カンマ (,)

### 戻り値

テキストファイルにエクスポートするデータの値を区切るのに使用される、デフォルトの文字列を制御します。OUTPUT 文に DELIMITED BY 句が含まれていない場合は、この設定値が使用されます。

### 参照


- 「isql_quote オプション [Interactive SQL]」896 ページ
- 「isql_escape_character オプション [Interactive SQL]」893 ページ
- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』


### 例

- 最初の例では、*c:¥Employees.txt* にエクスポートするデータのフィールドセパレーターとしてコロンを指定します。

```
SET OPTION isql_field_separator=':';
SELECT Surname, GivenName FROM Employees WHERE EmployeeID < 150;
OUTPUT TO c:¥Employees.txt FORMAT TEXT;
```

このコードを実行すると、次のデータが *Employees.txt* に書き込まれます。

```
'Whitney': 'Fran'
'Cobb':'Matthew'
'Chin':'Philip'
'Jordan':'Julie'
```

先頭と末尾の文字 (この場合は一重引用符) は、isql_quote の設定によって異なります。

- 次の例では、*c:¥Employees.txt* にエクスポートするデータのフィールドセパレーターとしてタブを指定します。

```
SET OPTION isql_field_separator='¥t';
SELECT Surname, GivenName FROM Employees WHERE EmployeeID < 150;
OUTPUT TO c:¥Employees.txt FORMAT TEXT;
```

このコードを実行すると、次のデータが *Employees.txt* に書き込まれます。

```
'Whitney' 'Fran'
'Cobb' 'Matthew'
'Chin' 'Philip'
'Jordan' 'Julie'
```

先頭と末尾の文字 (この場合は一重引用符) は、isql_quote の設定によって異なります。エスケープ文字 (この場合はバックスラッシュ) は、isql_escape_character の設定によって異なります。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## isql_maximum_displayed_rows オプション [Interactive SQL]

Interactive SQL で **[結果]** ウィンドウ枠に表示されるローの最大数を指定します。

### 指定可能な値

ALL または負でない整数

### デフォルト

500

### 戻り値

このオプションでは、**[結果]** ウィンドウ枠に表示されるローの最大数を指定できます。このオプションの値は、Interactive SQL の **[オプション]** ウィンドウでも設定できます。

> **警告**
> 表示する結果セットが大きい場合、Interactive SQL でメモリが不足することがあります。その場合、Interactive SQL から問題がレポートされ、結果セットは表示されません。

### 参照

- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』

## isql_print_result_set オプション [Interactive SQL]

*.sql* ファイルを実行したときに出力される結果セットを指定します。

### 指定可能な値

- **LAST**　ファイル内の最後の文の結果セットを出力します。
- **ALL**　結果セットを返すファイル内のすべての文の結果セットを出力します。
- **NONE**　結果セットを出力しません。

### デフォルト

LAST

### 戻り値

isql_print_result_set オプションは、Interactive SQL をコマンドラインプログラムとして実行する場合 (たとえば、*.sql* ファイルの実行時) のみ有効です。

このオプションでは、*.sql* ファイルを実行したときに出力する結果セットを指定できます。

### 参照

- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』

## isql_quote オプション [Interactive SQL]

テキストファイルにエクスポートするデータの文字列すべての先頭と最後に付く、デフォルトの文字列を制御します。

### 指定可能な値

文字列

### デフォルト

一重引用符 (')

### 備考

テキストファイルにエクスポートするデータの文字列すべての先頭と最後に付く、デフォルトの文字列を制御します。OUTPUT 文に QUOTE 句が含まれていない場合は、この設定値がデフォルトで使用されます。

### 参照

- 「isql_field_separator オプション [Interactive SQL]」 893 ページ
- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』

### 例

すべての文字列の先頭と最後に付くデフォルト文字列を二重引用符に変更するには、次のように指定します。

```
SET OPTION isql_quote="";
SELECT Surname, GivenName FROM Employees WHERE EmployeeID < 150;
OUTPUT TO c:¥Employees.txt FORMAT TEXT;
```

このコードを実行すると、次のデータが *Employees.txt* に書き込まれます。

```
"Whitney", "Fran"
"Cobb","Matthew"
"Chin","Philip"
"Jordan","Julie"
```

区切り文字 (この場合はカンマ) は、isql_field_separator の設定によって異なります。

## isql_show_multiple_result_sets オプション [Interactive SQL]

Interactive SQL で **[結果]** ウィンドウ枠に複数の結果セットを表示可能にするかどうかを指定します。

### 指定可能な値

On、Off

### デフォルト

Off

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 戻り値

複数の SELECT 文を返すプロシージャーを実行したときに複数の結果セットを **[結果]** ウィンドウ枠に表示するには、このオプションを ON に設定します。

各結果セットは **[結果]** ウィンドウ枠の個別のタブに表示されます。デフォルトでは、Interactive SQL では複数の結果セットは表示されません。このオプションの設定は、Interactive SQL をコマンドラインプログラムとして実行する場合にも適用されます。

Interactive SQL をウィンドウベースのアプリケーションとして実行している場合、isql_show_multiple_result_sets オプションは **[すべての結果セットを表示]** オプションとしても表示されます。「プロシージャーから複数の結果セットを返す」『SQL Anywhere サーバー SQL の使用法』を参照してください。

Interactive SQL を配備するときに、ユーザーによる isql_show_multiple_result_sets オプションの設定を禁止できます。showMultipleResultSets『SQL Anywhere サーバー プログラミング』を参照してください。

### 参照

- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』

### 例

次の例では、isql_show_multiple_result_sets を On に設定します。

```
SET OPTION isql_show_multiple_result_sets=On
```

## nulls オプション [Interactive SQL]

Interactive SQL で結果を表示するときにデータベース内の NULL 値をどのように表示するかを指定します。

### 指定可能な値

文字列

### デフォルト

(NULL)

### 備考

このオプションは好みに応じて設定します。結果セットをファイルに保存する場合は、このオプションの値は適用されません。結果セットをファイルに保存するときに使用される値は、output_nulls オプションで指定します。

### 参照

- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「output_nulls オプション [Interactive SQL]」900 ページ

## on_error オプション [Interactive SQL]

Interactive SQL の文を実行中にエラーが起こった場合の動作を制御します。

### 指定可能な値

- **Stop**　Interactive SQL が文の実行を停止します。
- **Prompt**　Interactive SQL は、ユーザーに続行するかどうかを確認するプロンプトを表示します。
- **Continue**　エラーは Interactive SQL の [メッセージ] タブに送信され、Interactive SQL は文の実行を続行します。
- **Exit**　Interactive SQL が停止します。
- **Notify_Continue**　エラーがレポートされ、ユーザーは続行するために [Enter] を押すか、**[OK]** をクリックするよう要求されます。
- **Notify_Stop**　エラーがレポートされ、ユーザーは実行を停止するために [Enter] キーを押すか、**[OK]** をクリックするよう要求されます。
- **Notify_Exit**　エラーがレポートされ、ユーザーは Interactive SQL を停止するために [Enter] キーを押すか、**[OK]** をクリックするよう要求されます。

### デフォルト

Prompt

### 戻り値

*.sql* ファイルを実行する場合は、Stop と Exit のどちらに設定しても同じ結果となります。これらの値のいずれかを指定すると、Interactive SQL が停止します。

Interactive SQL ユーティリティ (dbisql) の onerror オプションが指定されている場合、このオプションは無視されます。

### 参照


- 「Interactive SQL ユーティリティ (dbisql)」975 ページ
- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』


## output_format オプション [Interactive SQL]

ファイルにリダイレクトされる SELECT 文で検索したデータや、OUTPUT 文を使用した出力のデフォルトの出力フォーマットを設定します。

### 指定可能な値

- **TEXT**　出力は、ファイルの 1 行に 1 つのローが格納されたテキストフォーマットファイルです。すべての値をカンマで区切り、文字列をアポストロフィ (一重引用符) で囲みます。デ


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

リミターと引用符文字列は、DELIMITED BY と QUOTE 句を使って変更できます。QUOTE 句に All が指定されている場合は、文字列だけでなく、すべての値が引用符で囲まれます。

他の 3 つの特別なシーケンスも使用できます。2 つの文字 **¥n** は改行文字を表し、**¥¥** は 1 つの円記号 (¥) を表し、シーケンス **¥x***DD* は 16 進コード DD の文字を表します。

- **FIXED**　出力は、それぞれのカラムが固定幅を持つ固定フォーマットです。それぞれのカラムの幅は COLUMN WIDTH 句を使って指定できます。この句を省略した場合、各カラムの幅はカラムのデータ型から計算され、そのデータ型の値を保持するのに十分な大きさになります。カラムの見出しはこのフォーマットでは出力されません。
- **HTML**　出力は HTML フォーマットです。
- **SQL**　出力は、テーブル内の情報を再作成するのに必要な Interactive SQL の INPUT 文です。
- **XML**　出力は、UTF-8 でコード化され、DTD が埋め込まれた XML ファイルです。バイナリ値は、2 桁の 16 進数文字列として表されるバイナリデータとして CDATA ブロック内にコード化されます。

### 備考

output_format option は、OUTPUT 文で FORMAT 句が指定されない場合に使用されます。FORMAT 句が指定されている場合は、output_format オプションは無視されます。

### デフォルト

TEXT

### 参照

- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「OUTPUT 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』

## output_length オプション [Interactive SQL]

Interactive SQL が、外部ファイルに情報をエクスポートするときに使用するカラム値の長さを制御します。

### 指定可能な値

負でない整数

### デフォルト

0 (トランケーションなし)

### 戻り値

このオプションは、Interactive SQL が外部ファイルにデータをエクスポートするときに使用するカラム値の最大長を制御します (OUTPUT 文で出力リダイレクションを使用)。このオプションは、TEXT、HTML、SQL の出力フォーマットだけに影響します。

### 参照

- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』

## output_nulls オプション [Interactive SQL]

NULL 値をどのようにエクスポートするかを制御します。

### 指定可能な値

文字列

### デフォルト

空の文字列

### 戻り値

このオプションは、NULL 値をどのように OUTPUT 文で記述するかを制御します。結果セットで NULL 値が見つかった場合、NULL 値の代わりにこのオプションで設定された文字列が返されます。このオプションは、TEXT、HTML、FIXED、SQL の出力フォーマットだけに影響します。

### 参照

- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』

## truncation_length オプション [Interactive SQL]

表示内容を画面内に収めるために、幅の広いカラムのトランケーションを制御します。

### 指定可能な値

整数

### デフォルト

256

### 戻り値

truncation_length オプションは、表示されるカラムの長さを制限します。単位は文字です。値 0 はカラムがトランケートされないことを意味します。トランケーション長のデフォルト値は 256 です。

### 参照

- 「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

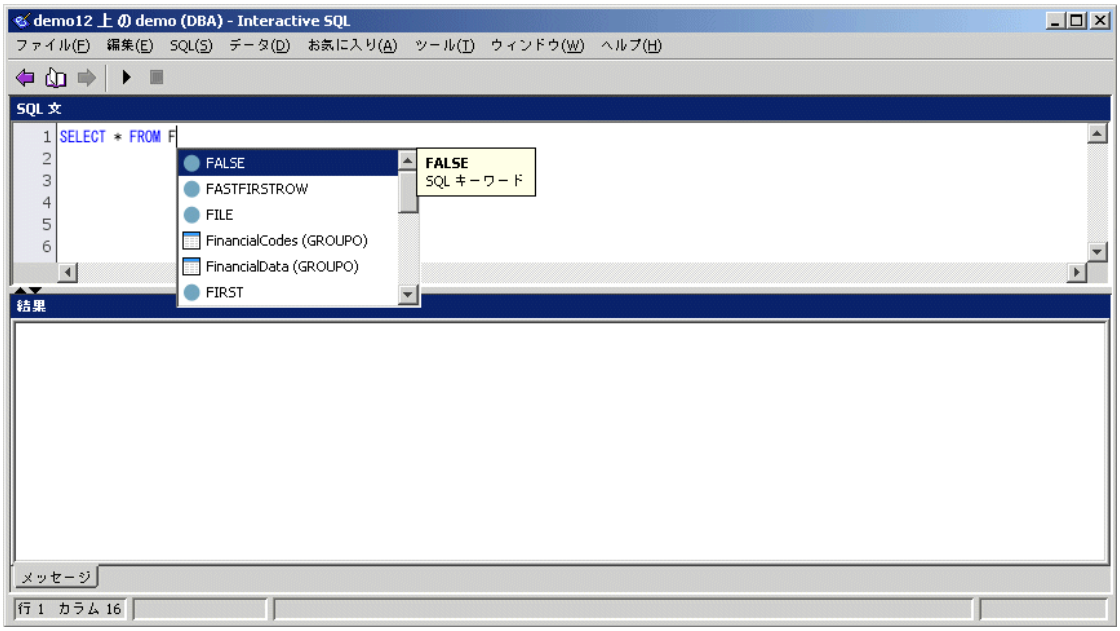
# テキスト補完の使用

Interactive SQL と Sybase Central では、入力時に完全なキーワードとオブジェクト名を供給可能なテキスト補完オプションが提供されます。たとえば、文字 S を入力すると、文字 S で始まる可能なキーワードまたはオブジェクト名のリストを含むテキスト補完ウィンドウが表示されます。テキスト補完設定は、Interactive SQL の **[オプション]** ウィンドウまたは Sybase Central のテキストエディターウィンドウから設定できます。

**♦ テキスト補完の使用**

1. Interactive SQL の **[SQL 文]** ウィンドウ枠で、データベースオブジェクト名の最初の文字を入力します。

2. デフォルトで、テキスト補完ウィンドウは入力すると自動的に表示されます。それ以外では、[Ctrl] キーとスペースキー、または [Ctrl + Shift] キーとスペースキーを押すと表示されます。テキスト補完ウィンドウには、入力した文字で始まるデータベースオブジェクトのリストが含まれます。次の例では、テキスト補完ウィンドウで文字 F で始まるすべてのデータベースオブジェクトが表示されます。




使用したいオブジェクト名が見つからない場合は、[Tab] キーを押して、データベースオブジェクトの完全なリストを表示します (完全なリストは、ユーザーが設定したフィルタリング オプションに基づいて表示されます。デフォルトでは、すべてのデータベースオブジェクトが表示されます)。

3. リストからオブジェクト名を選択し、[Enter] キーを押します。

   **[SQL 文]** ウィンドウ枠にオブジェクト名が表示されます。

## テキスト補完キーボードショートカット

テキスト補完リストが開いているときに、以下のキーボードショートカットを使用できます。

| キー | 説明 |
|---|---|
| [Ctrl + A] | 内容に関係なく一致するデータのリストを表示します。 |
| [Ctrl + 二重引用符 (")]<br><br>(または [Ctrl + Shift + ']) | quoted_identifier オプションの設定に関係なく、引用符で囲んで名前の入力を完了します。「quoted_identifier オプション」668 ページを参照してください。 |
| [Ctrl + アスタリスク (*)]<br><br>(または [Ctrl + Shift + 8]) | テーブルの場合、カラムのカンマ区切りのリスト (データ型を含む) を挿入します。<br><br>ストアドプロシージャーの場合、プロシージャー名を挿入し、続けてパラメーター名とデータ型のカンマ区切りのリストを挿入します。 |
| [Ctrl + C] | テキスト補完リストの内容を変更してカラムを表示、または非表示にします。 |
| [Ctrl + F] | テキスト補完リストの内容を変更して SQL 関数を表示、または非表示にします。 |
| [Ctrl + P] | テキスト補完リストの内容を変更してストアドプロシージャーおよび関数を表示、または非表示にします。 |
| [Ctrl + プラス記号 (+)]<br><br>(または [Ctrl + Shift + =]) | テーブルの場合、カラムのカンマ区切りのリストを挿入します。<br><br>ストアドプロシージャーの場合、プロシージャー名を挿入し、続けてパラメーター名のカンマ区切りのリストを挿入します。 |
| [Ctrl + S] | テキスト補完リストの内容を変更してシステムオブジェクトを表示、または非表示にします。 |
| [Ctrl + Shift] とスペースキー、または [Ctrl] キーとスペースキー | テキスト補完ウィンドウを開きます。[Ctrl] キーとスペースキーを使用して、テキスト補完ウィンドウを開くこともできます。 |
| [Ctrl + T] | テキスト補完リストの内容を変更してテーブルを表示、または非表示にします。 |
| [Ctrl + V] | テキスト補完リストの内容を変更してビューを表示、または非表示にします。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| キー | 説明 |
|---|---|
| [Esc] | テキストを追加しないでテキスト補完ウィンドウを閉じます。 |
| [Tab] | 選択内容を受け入れ、テキスト補完ウィンドウを閉じます。 |

# 高速ランチャーオプション

高速ランチャーオプションを使用して、Sybase Central および Interactive SQL の起動時間を短縮できます。高速起動を有効にすると、プログラムを閉じた後も、設定した一定期間はプログラムがメモリ内に残ります。この期間内にプログラムを再び起動すると、高速に起動します。期間内に再び起動しなかった場合は、プロセスが終了し、そのリソースがオペレーティングシステムに解放されます。

高速ランチャーオプションを有効にすると、システムトレイにプログラムのアイコンが表示されます。システムトレイアイコンを右クリックして、プログラムを再起動または停止できます。システムトレイアイコンを使用してプログラムを停止した場合、高速ランチャープロセスは終了し、そのリソースがオペレーティングシステムに解放されます。高速ランチャーは、Windows でのみ使用できます。

### 言語設定の変更

Sybase Central または Interactive SQL の停止後も高速ランチャーが実行されている場合は、dblang ユーティリティを使用して言語設定を変更しても、変更は Sybase Central が再起動されるまで有効になりません。

Windows で両方のユーティリティを停止するには、タスクマネージャーを使用してタスクを終了します。または、高速ランチャーの休止タイマーが切れるまで待ちます。

### 高速ランチャーオプションの設定

高速ランチャーオプションは、ユーザーのコンピューター上の TCP/IP ポートを使用します。別のプログラムがこのポートを使用している場合は、高速ランチャーによって使用されるポート番号を変更できます。

休止タイマーに指定した時間の間に高速ランチャーオプションが使用されない場合、高速ランチャーは終了し、他のアプリケーション用にメモリが解放されます。デフォルトでは、休止タイマーは 30 分に設定されています。

**♦ 高速ランチャーオプションを設定する (Interactive SQL および Sybase Central の場合)**

1. **[ツール]** » **[オプション]** をクリックします。
2. 左ウィンドウ枠で、**[一般]** をクリックします。
3. **[設定]** をクリックします。
4. **[ポート番号]** と **[高速ランチャーの停止]** のフィールドに入力します。

5. **[OK]** をクリックします。
6. **[OK]** をクリックします。

## 管理ツールの言語設定の変更

**♦ 高速ランチャーオプションが有効な場合に言語設定を変更する**

高速ランチャーオプションが有効になっている場合、言語設定に変更を加えても、現在のプロセスが停止されて再開されるまでは、Sybase Central または Interactive SQL によって検出されません。

1. **[ツール]** » **[オプション]** をクリックします。
2. **[オプション]** ウィンドウの **[一般]** タブで、**[高速ランチャーを有効にする]** オプションをオフにします。

   **[OK]** をクリックします。
3. Sybase Central または Interactive SQL を停止します。
4. 必要に応じて言語設定を変更します。たとえば、言語設定をドイツ語に変更するには次のコマンドを実行します。

   ```
   dblang DE
   ```
5. Sybase Central または Interactive SQL を起動します。
6. 高速ランチャーオプションを再び有効にします。
   a. **[ツール]** » **[オプション]** をクリックします。
   b. **[オプション]** ウィンドウの **[一般]** タブで、**[高速ランチャーを有効にする]** をクリックします。
   c. **[OK]** をクリックします。

**♦ 高速ランチャーオプションが無効な場合に言語設定を変更する**

1. Sybase Central または Interactive SQL を停止します。
2. 必要に応じて言語設定を変更します。たとえば、言語設定をドイツ語に変更するには次のコマンドを実行します。

   ```
   dblang de
   ```
3. Sybase Central または Interactive SQL を再起動します。

**参照**


- 「言語選択ユーティリティ (dblang)」983 ページ
- 「高速ランチャーオプション」903 ページ

# SQL Anywhere コンソールユーティリティ (dbconsole)

SQL Anywhere コンソールユーティリティ (dbconsole) は、データベースサーバー接続の管理機能とモニタリング機能を提供します。

dbconsole ユーティリティは、各種プラットフォームでサポートされています。利用可能なプラットフォームについては、http://www.ianywhere.jp/tech/1061806-os_components.html を参照してください。

dbconsole コンソールユーティリティがサポートされていないプラットフォームでは、接続、データベース、データベースサーバーのプロパティを使用して情報を取得したり、SQL Anywhere コンソールユーティリティをサポートするオペレーティングシステム (Windows、Mac OS X、Linux など) を実行するコンピューターからデータベースサーバーをモニタリングすることができます。

DBA 権限を持たないユーザーが SQL Anywhere Console ユーティリティに接続すると、DBA 権限を必要とするすべての機能は無効になります。

SQL Anywhere コンソールユーティリティでサポートされるオプションの詳細については、「SQL Anywhere コンソールユーティリティ (dbconsole)」1021 ページを参照してください。

## SQL Anywhere コンソールユーティリティ (dbconsole) の起動

**♦ SQL Anywhere コンソールユーティリティを起動する (コマンドラインの場合)**

● 次のコマンドを実行します。

```
dbconsole
```

データベースへの接続パラメーターを指定する -c オプションを省略するなど、接続パラメーターの指定が不十分であると、**[接続]** ウィンドウが表示されます。このウィンドウで、データベースへの接続情報を入力できます。

サポートされるオプションの詳細については、「SQL Anywhere コンソールユーティリティ (dbconsole)」1021 ページを参照してください。

次のコマンドは、SQL Anywhere コンソールユーティリティを起動し、サンプルデータベースに接続します。

```
dbconsole -c "UID=DBA;PWD=sql;DSN=SQL Anywhere 12 Demo"
```

**♦ SQL Anywhere コンソールユーティリティを起動する (Linux デスクトップアイコンの場合)**

Linux のデスクトップアイコンをサポートする Linux バージョンを使用していて、SQL Anywhere のインストール時にこれらのアイコンをインストールするように選択した場合は、次の手順を使用できます。

1. **[アプリケーション]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[SQL Anywhere コンソール]** をクリックします。

2. **[接続]** ウィンドウで、データベースの接続情報を入力します。
3. **[接続]** をクリックします。

**♦ SQL Anywhere コンソールユーティリティを起動する (UNIX コマンドラインの場合)**

> **注意**
> 以降の手順は、SQL Anywhere ユーティリティのソースを指定済みであることを前提としています。「Unix および Mac OS X 環境変数」443 ページを参照してください。

1. ターミナルセッションで次のコマンドを実行します。

```
dbconsole
```

2. **[接続]** ウィンドウで、データベースの接続情報を入力します。
3. **[接続]** をクリックします。

**♦ SQL Anywhere コンソールユーティリティを起動する (Mac OS X の場合)**

> **注意**
> 管理ツールは、Apple JDK 1.6 (Mac OS X 10.5.2 以降) でサポートされている、64 ビットのプロセッサーを搭載した Intel ベースの Macintosh コンピューターだけで動作します。http://www.ianywhere.jp/tech/1061806-os_components.html を参照してください。

1. Finder で、*/Applications/SQLAnywhere* にある [DBConsole] をダブルクリックします。12
2. **[接続]** ウィンドウで、データベースの接続情報を入力します。
3. **[接続]** をクリックします。

**参照**

- 「[接続] ウィンドウを開く (Sybase Central の場合)」14 ページ

## SQL Anywhere コンソールユーティリティ (dbconsole) のメインウィンドウのナビゲーション

dbconsole ユーティリティは 3 つのウィンドウ枠で構成されています。

- **[接続]**　現在のデータベース接続に関する情報を表示します。
- **[プロパティ]**　現在実行中のデータベースとデータベースサーバーに関する情報を表示します。
- **[メッセージ]**　データベースサーバーメッセージを表示します。

**[オプション]** ウィンドウを使用して、各ウィンドウ枠に表示される情報を設定できます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**♦ [接続] ウィンドウ枠の内容をカスタマイズする**

1. dbconsole ユーティリティで、**[ファイル]** » **[オプション]** をクリックします。
2. 左ウィンドウ枠で、**[接続ビューアー]** をクリックします。
3. **[接続]** ウィンドウ枠に表示するプロパティを選択します。
4. **[OK]** をクリックします。

**♦ [プロパティ] ウィンドウ枠の内容をカスタマイズする**

1. dbconsole ユーティリティで、**[ファイル]** » **[オプション]** をクリックします。
2. 左ウィンドウ枠で、**[プロパティビューアー]** をクリックします。
3. **[プロパティ]** ウィンドウ枠に表示するデータベースとデータベースサーバーのプロパティを選択します。
4. **[OK]** をクリックします。

**♦ [メッセージ] ウィンドウ枠の内容をカスタマイズする**

1. dbconsole ユーティリティで、**[ファイル]** » **[オプション]** をクリックします。
2. 左ウィンドウ枠で、**[メッセージビューアー]** をクリックします。
3. **[メッセージ]** ウィンドウ枠に表示するメッセージのメッセージオプションを選択します。
4. **[OK]** をクリックします。

# ソフトウェアの更新

EBF などの更新やメンテナンスリリースが使用可能になったときに通知されるように SQL Anywhere を設定できます。EBF とメンテナンスリリースの詳細については、-iu オプション、dbsupport1029 ページを参照してください。

デフォルトでは、SQL Anywhere はソフトウェア更新を毎日チェックします。

## 更新の自動チェック

Sybase Central、Interactive SQL、SQL Anywhere コンソールユーティリティ (dbconsole) には、SQL Anywhere によるソフトウェア更新チェックの有無とその頻度を制御する、更新チェッカーを設定する手段が用意されています。

**♦ 更新チェッカーを設定する (Sybase Central の場合)**

1. **[ヘルプ]** » **[SQL Anywhere 12]** » **[更新チェッカーの設定]** をクリックします。
2. 更新チェッカーの設定を編集します。

3. **[OK]** をクリックします。

♦ **更新チェッカーを設定する (Interactive SQL の場合)**

1. **[ツール]** » **[オプション]** をクリックします。
2. 左ウィンドウ枠で、**[SQL Anywhere]** をクリックします。
3. **[更新チェッカー]** タブをクリックします。
4. 更新チェッカーの設定を編集します。
5. **[OK]** をクリックします。

♦ **更新チェッカーを設定する (SQL Anywhere コンソールユーティリティの場合)**

1. **[ファイル]** » **[オプション]** をクリックします。
2. 左ウィンドウ枠で、**[更新チェッカー]** をクリックします。
3. 更新チェッカーの設定を編集します。
4. **[OK]** をクリックします。

### 更新の手動チェック

SQL Anywhere ソフトウェアの更新は、次のいずれかの方法でいつでもチェックできます。

- **[スタート] メニュー**　**[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[更新のチェック]** をクリックします。
- **Sybase Central**　**[ヘルプ]** » **[SQL Anywhere 12]** » **[更新のチェック]** をクリックします。
- **Interactive SQL**　**[ヘルプ]** » **[更新のチェック]** をクリックします。
- **SQL Anywhere コンソールユーティリティ (dbconsole)**　**[ヘルプ]** » **[更新のチェック]** をクリックします。
- **SQL Anywhere サポートユーティリティ (dbsupport)**　次のコマンドを実行します。

  ```
  dbsupport -iu
  ```

- **Sybase の Web サイト**　http://downloads.sybase.com に移動してください。このページを表示するには、Sybase.com アカウントが必要です。

### 参照


- 「SQL Anywhere のエラーレポート」1210 ページ
- 「サポートユーティリティ (dbsupport)」1027 ページ

# データベース管理ユーティリティ

SQL Anywhere には、データベース管理タスクを行う一連のユーティリティプログラムが付属しています。各ユーティリティには、Sybase Central または Interactive SQL からアクセスするか、コマンドプロンプトでアクセスできます。

利用可能なプラットフォームについては、http://www.ianywhere.jp/sas/os.html を参照してください。

管理ユーティリティでは、一連のレジストリエントリまたは *.ini* ファイルを使用します。「レジストリと INI ファイル」463 ページを参照してください。

**参照**


- 「Sybase Central」815 ページ
- 「Interactive SQL」840 ページ


# 設定ファイル

SQL Anywhere 12 で提供されている多くのユーティリティでは、設定ファイルにコマンドラインオプションを保存できます。オプションの拡張セットを使用する場合は、それらを設定ファイルに保存すると便利です。

*@data* オプションを使用すると、コマンドラインで環境変数と設定ファイルを指定できます。設定ファイルを指定するには、*data* を設定ファイルのパスと名前で置き換えます。同じ名前の環境変数と設定ファイルが存在する場合は、環境変数が使用されます。

設定ファイルには、改行を含めたり、@data オプションを含むあらゆるオプションの設定を格納したりできます。行の先頭にシャープ記号 (#) を使用すると、その行をコメントとして指定することができます。行の最後に単独で存在するアンパサンド (&) 文字は、前のトークンが次の行に続くことを示します。たとえば、次の設定ファイルを使用して、強力な暗号化を許可するサーバーを起動できます。

```
-ec TLS(TLS_TYPE=RSA;FIPS=Y;IDENTITY=rsaserver.id; &
    IDENTITY_PASSWORD=test)
 -x tcpip c:¥mydemo.db
```

*@data* パラメーターはコマンドラインの任意の位置に指定でき、ファイルに含まれるパラメーターがその位置に挿入されます。さらに *@data* は、1 つのコマンドラインで複数回使用して複数の設定ファイルを指定できます。

ユーティリティは、指定の設定ファイルを展開し、コマンドライン全体を左から右へ読み取ります。コマンドライン内のその他のオプションで上書きされるオプションを指定した場合、行の終端に近い方のオプションが有効になります。競合するオプションのためにエラーが発生することがあります。

**注意**
サーバーバックグラウンド起動ユーティリティ (dbspawn) では、*@data* オプションで指定された設定ファイルは展開されません。

設定ファイル内のパスワードなどの情報を保護する場合は、ファイル非表示ユーティリティを使用して、設定ファイルの内容を読みにくくすることができます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。

### 設定ファイルのエスケープ文字

設定ファイルでは、オプションの値にスペースを含む値が含まれている場合は、その値を二重引用符で囲む必要があります。さらに、引用符で囲んだ値にスペースを含む別の値が含まれている場合は、その別の値を囲む二重引用符をエスケープする必要があります。SQL Anywhere では、¥ のエスケープシーケンスとして ¥¥、" のエスケープシーケンスとして ¥" がサポートされています。

たとえば、次に示す dblsn 設定ファイルの抜粋には、エスケープされた二重引用符が含まれています。

```
-l "subject=$remote_id;
   content=sync cardealer;
   action='run dbmlsync.exe -c ¥"filedsn=c:¥my fdsns¥CarDealer.dsn¥"
   -ot dbmlsync.log -k -e sa=on';"
```

### 例

次の設定ファイルには、検証ユーティリティ (dbvalid) の一連のオプションが含まれています。

```
#Connect to the sample database as the user DBA with password sql
-c "UID=DBA;PWD=sql;DBF=%SQLANYSAMP12%¥demo.db"
#Perform an express check on each table
-fx
#Log output messages to the specified file
-o "c:¥validationlog.txt"
```

この設定ファイルを *c:¥config.txt* として保存すると、コマンドで次のように使用できます。

```
dbvalid @c:¥config.txt
```

## 設定ファイルでの条件付き解析

設定ファイルで条件付き解析を使用することによって、そのファイルを使用できるユーティリティを指定できます。条件ディレクティブを使用すると、設定ファイルを使用するユーティリティに応じてコマンドパラメーターを適用したり除外したりすることができます。設定ファイルで条件付き解析を使用する場合でも、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容を非表示にすることができます。

### 構文

*configuration-file= text...*

*text* : *comment* | *conditional* | *command-line-option*

*comment* : *line starting with # that is not a conditional*

*conditional* :

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**#if** *condition*
*text*
[ **#elif** *condition*
*text*
] ...
[ **#else**
*text*
] ...
**#endif**

*condition* : { **tool=***utility-name*[,*utility-name*]... | *utility-name* }

*utility-name* では、次の値がサポートされています。

| | | | | |
|---|---|---|---|---|
| dbbackup | dbinfo | dbmlsync | dbstop | dbxtract |
| dbdsn | dbinit | dbping | dbsupport | mlsrv |
| dbeng | dblic | dbremote | dbsvc | mluser |
| dberase | dblocate | dbspawn | dbunload | qaagent |
| dbfhide | dblog | dbsrv | dbupgrad | mlstop |
| dbhist | dblsn | dbstats | dbvalid | |

## 備考

ディレクティブと認識されるようにするには、1 つの行で空白以外の最初の文字を # とします。#if または #elif ディレクティブ内にユーティリティが含まれている場合、そのディレクティブから次の条件付きディレクティブまでの行が処理の対象となります。#else ディレクティブは、それより前のブロックにユーティリティがない場合に実行する処理を示します。条件付きディレクティブ構造は、#endif ディレクティブで終了します。

**tool=** で指定するツール名のリスト内に空白を入れることはできません。条件付きディレクティブはネストすることができます。設定ファイルの解析中にエラーが発生すると、ユーティリティは設定ファイルを開くことができない旨をレポートします。

dbspawn ユーティリティでは、作成するコマンドで設定ファイル参照を指定できますが、dbspawn ユーティリティのオプションを使用して設定ファイルを指定することはできません。たとえば、次の最初のコマンドはサポートされていますが、2 つ目のコマンドはサポート**されていません**。

```
dbspawn dbeng12 @myconfig.ini
```

```
dbspawn @spawnopts.ini dbeng12 demo.db
```

### 参照


- 「バックアップユーティリティ (dbbackup)」913 ページ
- 「データソースユーティリティ (dbdsn)」936 ページ
- 「@data dbeng12/dbsrv12 サーバーオプション」201 ページ
- 「消去ユーティリティ (dberase)」951 ページ
- 「ファイル非表示ユーティリティ (dbfhide)」953 ページ
- 「ヒストグラムユーティリティ (dbhist)」955 ページ
- 「情報ユーティリティ (dbinfo)」957 ページ
- 「初期化ユーティリティ (dbinit)」958 ページ
- 「サーバーライセンス取得ユーティリティ (dblic)」1003 ページ
- 「サーバー列挙ユーティリティ (dblocate)」999 ページ
- 「トランザクションログユーティリティ (dblog)」1041 ページ
- 「Windows デバイス用の Mobile Link Listener ユーティリティ (dblsn)」『Mobile Link サーバー起動同期』
- 「dbmlsync 構文」『Mobile Link クライアント管理』
- 「Ping ユーティリティ (dbping)」994 ページ
- 「SQL Remote Message Agent ユーティリティ (dbremote)」『SQL Remote』
- 「サーバーバックグラウンド起動ユーティリティ (dbspawn)」1022 ページ
- 「サーバー停止ユーティリティ (dbstop)」1024 ページ
- 「Linux 用サービスユーティリティ (dbsvc)」1007 ページ
- 「Windows 用サービスユーティリティ (dbsvc)」1013 ページ
- 「アンロードユーティリティ (dbunload)」1044 ページ
- 「アップグレードユーティリティ (dbupgrad)」1061 ページ
- 「検証ユーティリティ (dbvalid)」1064 ページ
- 「抽出ユーティリティ (dbxtract)」『SQL Remote』
- 「mlsrv12 構文」『Mobile Link サーバー管理』
- 「Mobile Link ユーザー認証ユーティリティ (mluser)」『Mobile Link サーバー管理』
- 「qaagent ユーティリティ」『QAnywhere』


### 例

次の設定ファイルを使用できるユーティリティは、dbping、dbstop、dbvalid です。

```
#if tool=dbping,dbstop,dbvalid
  #always make tools quiet
  -q
  -c "UID=DBA;PWD=sql;Host=myhost;DBN=mydb"
  #if dbping
    #make a database connection
    -d
  #elif tool=dbstop
    #don't ask
    -y
  #else
    #must be dbvalid
    #use WITH EXPRESS CHECK
    -fx
  #endif
#endif
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

# バックアップユーティリティ (dbbackup)

実行中のデータベースのデータベースファイルやトランザクションログについて、クライアント側またはサーバー側のバックアップを作成します。

## 構文

**dbbackup** [ *options* ] *target-directory*

| オプション | 説明 |
|---|---|
| *@data* | 指定された環境変数または設定ファイルからオプションを読み込みます。「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル難読化ユーティリティを使用して、設定ファイルの内容を難読化できます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |
| **-b** *block-size* | データベースサーバーから dbbackup へのページ転送に使用するブロックの最大サイズ (ページ数) を指定します。dbbackup ユーティリティは、指定された数のページを割り付けようとします。割り付けが失敗した場合は、この値を半分にして再試行し、成功するまでこれを繰り返します。デフォルトのサイズは 128 です。 |
| **-c** *"keyword=value; ..."* | 接続パラメーターを指定します。データベースに接続するためには、DBA 権限または REMOTE DBA 権限 (SQL Remote) のあるユーザー ID を使用する必要があります。「接続パラメーター」320 ページを参照してください。 |
| **-d** | トランザクションログファイルがあっても、それをバックアップしないで、メインデータベースファイルだけをバックアップします。 |

| オプション | 説明 |
| --- | --- |
| **-k** *checkpoint-log-copy-option* | バックアップ先のディレクトリに書き込む前に dbbackup でデータベースファイルに対してどのような処理を実行するかを指定します。バックアップ中に更新前イメージを適用するか、チェックポイントログをバックアップとしてコピーするかを選択します。どちらを選択するかによって、パフォーマンスに違いが生じます。-s オプションを指定してサーバー側でバックアップを実行する場合、-k のデフォルト設定は auto です。それ以外の場合は、copy がデフォルト設定となります。<br><br>**auto**　データベースサーバーはバックアップディレクトリがあるボリュームの空きディスク領域サイズをチェックします。バックアップを開始する時点でデータベースサイズの 2 倍以上の空きディスク領域がある場合は、copy を指定した場合と同じ方法でバックアップが実行されます。それ以外の場合は、nocopy を指定した場合と同じ方法でバックアップが実行されます。この設定は、-s を指定した場合のみ使用できます。<br><br>**copy**　バックアップでは、変更されたページの更新前イメージを適用せずにデータベースファイルを読み込みます。チェックポイントログの全体とシステム DB 領域がバックアップディレクトリにコピーされます。このデータベースを次に起動すると、データベースは自動的にバックアップ開始時のチェックポイントの状態にリカバリされます。<br><br>このオプションを使用すると、ページの更新前イメージをテンポラリファイルに書き込む必要がないため、バックアップパフォーマンスが向上し、バックアップ中に動作中の他の接続との内部サーバー競合が減少します。ただし、データベースファイルのバックアップコピーにはチェックポイントログが含まれ、このログにはバックアップの開始以降に変更されたページの更新前イメージが保存されているため、バックアップコピーのサイズがバックアップを開始した時点のデータベースファイルより大きくなることがあります。copy オプションは、バックアップ先ディレク |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| | トリのディスク領域が十分である場合に使用してください。<br><br>**nocopy**　チェックポイントログがバックアップとしてコピーされません。この場合、変更されたページの更新前イメージはテンポラリファイルに保存され、バックアップの実行中はバックアップに適用されます。データベースファイルのバックアップコピーは、バックアップを開始した時点のデータベースと同じサイズになります。バックアップコピーは、チェックポイントログが含まれていないため、実際には多少小さくなることがあります。このオプションを指定すると、データベースファイルのバックアップは小さくなりますが、バックアップの速度が遅くなり、データベースサーバーで実行される他の操作のパフォーマンスが低下することがあります。バックアップ先のドライブの空き領域が少ない場合に、このオプションが役に立ちます。<br><br>**recover**　copy オプションを指定した場合と同じようにチェックポイントログがコピーされますが、このチェックポイントログはバックアップの完了時にデータベースに適用されます。したがって、データベースファイルのバックアップは、バックアップ操作を開始した時点と同じ状態 (および同じサイズ) となります。このオプションが役に立つのは、バックアップドライブの空き領域が少ない場合です (チェックポイントログをバックアップする copy オプションの場合と同じ量の空き領域が必要ですが、バックアップファイルのサイズは copy オプションを指定した場合より小さくなります)。この設定は、-s を指定した場合のみ使用できます。 |

| オプション | 説明 |
| --- | --- |
| **-l** *filename* | サーバーがクラッシュした場合に第 2 のシステムをすばやく起動できるようにします。サーバーが実行されている間、ライブバックアップは停止しないで実行を続けます。ライブバックアップは、プライマリサーバーが使用できなくなるまで実行されます。クラッシュが発生した時点でライブバックアップは停止しますが、バックアップされたログファイルはそのまま残り、第 2 のシステムを即座に起動するために使用できます。「ライブバックアップとトランザクションログミラーの違い」1076 ページと「ライブバックアップの作成」1086 ページを参照してください。<br><br>-l を指定する場合は、-s を使用してサーバー上にバックアップを作成できません。 |
| **-n** | バックアップトランザクションログファイルの命名規則を *yymmddxx.log* に変更します。*xx* は **AA** から **ZZ** までの連続した英字を表し、*yymmdd* は現在の年月日を表します。このオプションは -r とともに使用します。<br><br>トランザクションログファイルのバックアップコピーは、*yymmddxx.log* の命名規則に基づいて、コマンドで指定したディレクトリに保存されます。これにより、トランザクションログファイルの複数のバージョンのバックアップを、同じバックアップディレクトリに保存することができます。<br><br>また、-x オプションと -n オプションの両方を使用して、ログコピーの名前を変更することもできます。次に例を示します。<br><br>dbbackup -c "UID=DBA;PWD=sql" -x -n mybackupdir |
| **-o** *filename* | 指定したファイルに、出力メッセージを書き込みます。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| **-p** | データベースサーバーからクライアントに、フォーマットされた進行メッセージを送信します。<br><br>-s オプションを使用してサーバー側のバックアップを実行しないかぎり、-p オプションは無視されます。<br><br>フォーマットされた進行メッセージの詳細については、「progress_messages オプション」 665 ページ を参照してください。 |
| **-q** | 出力メッセージを表示しません。このオプションは、このユーティリティをコマンドプロンプトで実行する場合のみ使用できます。 |
| **-r** | トランザクションログの名前を変更して新しいログを開始します。このオプションを使用すると、チェックポイントが発生し、次の 3 つの手順が発生します。<br><br>1. 現在作業しているトランザクションログファイルのコピーが作成され、コマンドで指定したディレクトリに保存されます。<br><br>2. 現在のトランザクションログは現在のディレクトリ内に残りますが、*yymmddxx.log* のフォーマットに従って名前が変更されます。*xx* は *AA* から *ZZ* までの連続した英字を表し、*yymmdd* は現在の年月日を表します。このファイルは、現在のトランザクションログではなくなります。<br><br>3. トランザクションを含まない新しいトランザクションログファイルが作成されます。このファイルに、直前まで現在のトランザクションログとして使用されていたファイルの名前が付けられ、データベースサーバーによって現在のトランザクションログとして使用されます。 |

| オプション | 説明 |
| --- | --- |
| **-s** | BACKUP DATABASE 文を使用して、サーバー上にイメージのバックアップを作成します。-s オプションを指定する場合、-l オプション (トランザクションログのライブバックアップを作成) は使用できません。指定されたディレクトリは、サーバーの現在のディレクトリに対する相対パスなので、完全パス名を指定することをおすすめします。さらに、サーバーには指定されたディレクトリへの書き込みパーミッションが必要です。-s を指定すると、バックアップユーティリティは進行状況のメッセージを表示せず、既存のファイルを上書きするときにプロンプトを表示しません。既存のファイルを上書きするときにプロンプトを表示させる場合は、-s または -y を指定しないでください。-k リカバリオプションを指定する場合は、-s オプションも必ず指定してください。 |
| **-t** | トランザクションログをデータベースファイルの最新のバックアップコピーに対して適用できるので、インクリメンタルバックアップとして使用できるバックアップを作成します。 |
| **-x** | 既存のトランザクションログをバックアップし、元のログを削除して、新しいトランザクションログを開始します。データベースミラーリングを使用している場合は、このオプションを使用しないでください。「データベースミラーリングとトランザクションログファイル」1172 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| **-xo** | 現在のトランザクションログを削除して、新しいトランザクションログを開始します。この操作では、バックアップは実行されません。この操作の目的は、レプリケーション環境以外の環境でディスク領域を解放することです。データベースミラーリングを使用している場合は、このオプションを使用しないでください。「データベースミラーリングとトランザクションログファイル」1172 ページを参照してください。<br><br>**警告**<br>このオプションを使用すると、データベースがメディア障害からリカバリできなくなる可能性があります。このオプションは、データの損失を許容できる場合にのみ使用してください。<br><br>レプリケートされるデータベースまたは同期されるデータベースでは、-xo オプションを使わないでください。SQL Remote と Mobile Link はトランザクションログ情報に依存します。 |
| **-y** | 確認メッセージを表示することなく、バックアップディレクトリを作成するか、ディレクトリ内の既存のバックアップファイルを置き換えます。既存のファイルを上書きするときにプロンプトを表示させる場合は、-s または -y を指定しないでください。 |
| *target-directory* | バックアップファイルのコピー先ディレクトリを指定します。このディレクトリが存在しない場合は作成されます。ただし、親ディレクトリが存在していなければなりません。デフォルトでは、バックアップユーティリティは、データベースファイルのクライアント側のバックアップを作成します。-s オプションを指定すると、BACKUP DATABASE 文を使用してサーバー上にバックアップを作成できます。 |

## 備考

バックアップユーティリティを使うと、すべてのバックアップコピーを単一データベースに作成することができます。単純なデータベースは、メインデータベースファイルとトランザクションログの 2 つのファイルからなります。より複雑なデータベースは、複数のファイルにテーブルを格納できます。各ファイルは個別の DB 領域となります。すべてのバックアップデータベースファイル名はデータベースファイル名と同じです。バックアップユーティリティで作成したイメージのバックアップは、バックアップされた各ファイルの別ファイルで構成されます。

アーカイブバックアップ (データベースファイルとトランザクションログを含む 1 つのファイル) 作成の詳細については、「アーカイブバックアップ」1077 ページを参照してください。

実行中のデータベースに対してバックアップユーティリティを使用するのは、データベースが実行されていないときにデータベースファイルをコピーするのと同じです。バックアップユーティリティを使用すると、他のアプリケーションやユーザーがデータベースを使用しているときでも、そのデータベースをバックアップできます。

-d または -t のいずれのオプションも使用されていない場合は、すべてのデータベースファイルがバックアップされます。

デフォルトでは、バックアップユーティリティは、データベースファイルのクライアント側のバックアップを作成します。-s オプションを指定すると、BACKUP DATABASE 文を使用してサーバー上にバックアップを作成できます。

サーバー側でのバックアップ実行の詳細については、「BACKUP 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

**警告**
データベースとトランザクションログのバックアップコピーには、どのような変更でも加えるべきではありません。バックアップ中に処理中のトランザクションがなかった場合、または BACKUP DATABASE WITH CHECKPOINT LOG RECOVER か WITH CHECKPOINT LOG NO COPY を指定した場合は、読み込み専用モードを使用するか、バックアップデータベースのコピーを検証して、バックアップデータベースの妥当性をチェックできます。

一方、トランザクションの処理中だった場合、または BACKUP DATABASE WITH CHECKPOINT LOG COPY を指定した場合は、検証の開始時にデータベースサーバーがデータベースのリカバリを実行する必要が生じます。リカバリを実行するとバックアップコピーに変更が加えられますが、これは望ましいことではありません。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

終了コードの詳細については、「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』を参照してください。

## 参照

dbbackup のほかに、次の方法を使用して、バックアップユーティリティにアクセスできます。

- Sybase Central の**バックアップイメージ作成ウィザード**を使用する。「イメージバックアップ」1077 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

●Interactive SQL の BACKUP DATABASE 文を使用する。「BACKUP 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

推奨されるバックアップ手順の詳細については、「バックアップとデータリカバリ」1071 ページを参照してください。

### 例

たとえば、次のコマンドは、DBA ユーザーとして sample_host という名前のコンピューターに接続し、このコンピューターで実行中のサンプルデータベースを *SQLAnybackup* ディレクトリにバックアップします。

```
dbbackup -c "Host=sample_host;DBN=demo;UID=DBA;PWD=sql" SQLAnybackup
```

## Broadcast Repeater ユーティリティ (dbns12)

SQL Anywhere クライアントは、UDP ブロードキャストは通常到達しないような、別のサブネット上やファイアウォールを介して実行している SQL Anywhere データベースサーバーを検索できます。

### 構文

**dbns12** [ *options* ] [ *address* ... ]

| オプション | 説明 |
|---|---|
| *@data* | 指定された環境変数または設定ファイルからオプションを読み込みます。「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル非表示ユーティリティを使用して、設定ファイルの内容を読みにくくすることができます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |
| **-ap** *port* | データベースサーバーで使用されるポート番号を指定します。デフォルトのポート番号は 2638 です。 |
| **-m** *ip* | DBNS プロセスが実行されているコンピューターの IP アドレスを指定します。このパラメーターは、複数の IP アドレスがあるコンピューターでは必須です。このアドレスは IPv4 アドレスであることが必要です。 |

| オプション | 説明 |
| --- | --- |
| **-o** *filename* | Broadcast Repeater のメッセージウィンドウに表示される出力を指定されたファイルに書き込みます。 |
| **-p** *port* | DBNS Broadcast Repeater で使用されるポート番号を指定します。デフォルトは 3968 です。サブネット間にファイアウォールがある場合は、標準のクライアント／サーバー通信用のポート 2638 に加え、Broadcast Repeater ユーティリティが DBNS プロセス間の TCP 接続に使用するポートも開く必要があります。 |
| **-q** | クワイエットモードで実行します (メッセージを表示しません)。 |
| **-s** | 新しい DBNS プロセスは別の DBNS プロセスが同じサブネット上で実行されていないかどうかをチェックし、既存の DBNS プロセスが見つかった場合は、エラーを返してから停止します。 |
| **-ud** | UNIX オペレーティングシステムで、DBNS Broadcast Repeater をデーモンとして実行します。DBNS Broadcast Repeater をデーモンとして実行する場合は、-o オプションを指定して、出力情報のログを取ってください。<br><br>DBNS Broadcast Repeater をデーモンとして起動すると、現在のユーザーの umask 設定によってパーミッションが制御されます。DBNS Broadcast Repeater に適切なパーミッションを付与するため、DBNS Broadcast Repeater を起動する前に umask 値を設定することをおすすめします。 |
| **-ui** | X Window サーバーがサポートされている Linux で、使用可能な表示がない場合にシェルモードで DBNS Broadcast Repeater を起動します。 |
| **-ux** | Linux で **[SQL Anywhere Broadcast Repeater]** ウィンドウを開きます (X-Window Server を使用)。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| **-x** *host* | 指定するホストで実行中の DBNS プロセスを停止します。IP アドレスとホスト名のどちらでも指定できます。 |
| **-z** | DBNS Broadcast Repeater をデバッグモードで起動します。デバッグモードで実行すると、受信または送信される各 SQL Anywhere ブロードキャストパケットに対応する行が Broadcast Repeater のメッセージウィンドウに表示されます。デバッグ出力が冗長であるため、接続上の問題がある場合のみデバッグモードを使用してください。 |
| *address* | DBNS プロセスを実行中または将来実行する他のコンピューターの IP アドレスまたはホスト名を指定します。これによって、DBNS プロセスが互いを検出し、既知のデータベースサーバーや他の DBNS プロセスの情報を交換できます。 |

## 備考

Broadcast Repeater を使用すると、UDP ブロードキャストは通常到達しない、別のサブネット上やファイアウォール越しの SQL Anywhere データベースサーバーであっても、SQL Anywhere クライアントは HOST 接続パラメーターや LDAP を使用することなく検索できます。

*address* には、IP アドレスとコンピューター名のどちらでも指定できます。複数のアドレスを指定する場合は、スペースで区切ります。

このユーティリティは、サポートされている UNIX プラットフォームと、32 ビットおよび 64 ビットのすべての Windows プラットフォームで使用できます。

Broadcast Repeater を使用するためには、クライアントとデータベースサーバーで SQL Anywhere 9.0.2 以降が実行されている必要があります。

**警告**
dbns12 ユーティリティを SQL Anywhere データベースサーバーと同じコンピューター上で実行しないようおすすめします。dbns12 またはデータベースサーバーが UDP ブロードキャストを受信しなくなる可能性があります。

## 参照

- 「Broadcast Repeater ユーティリティを使用したデータベースサーバーの検出」82 ページ

## 例

サブネット 10.50.83.255 と 10.50.125.255 のコンピューターがブロードキャストを使用して接続できるようにする場合を想定します。10.50.83.255 サブネット上のコンピューター (10.50.83.114

のコンピューター A) と 10.50.125.255 サブネット上のコンピューター (10.50.125.103 のコンピューター B) が必要です。

両方のコンピューターで dbns12 を実行し、それぞれ他方のコンピューターの IP アドレスを渡します。コンピューター A では次のコマンドを実行します。

```
dbns12 10.50.125.103
```

コンピューター B では次のコマンドを実行します。

```
dbns12 10.50.83.114
```

いずれかのコンピューターに複数の IP アドレスがある場合は、-m オプションを使用してローカル IP アドレスを指定する必要があります。

コンピューター A では次のコマンドを実行します。

```
dbns12 -m 10.50.83.114  10.50.125.103
```

# 証明書作成ユーティリティ (createcert)

X.509 証明書を作成します。

### 構文

**createcert [ -r | -s ]**

| オプション | プロンプト名 | 説明 |
|---|---|---|
| *@data* | N/A | 指定された環境変数または設定ファイルからオプションを読み込みます。「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容を難読化できます。 「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |

| オプション | プロンプト名 | 説明 |
| --- | --- | --- |
| **-b** *value* | **[RSA のキー長を入力してください (512 ～ 16384)。]** | RSA キー長を指定します。512 ～ 16384 ビットの間で長さを選択できます。<br><br>このオプションは、-r オプションが指定されており、-t オプションに RSA が指定されている場合にのみ必要です。 |
| **-c** *filename* | **[署名者の証明書のファイルパスを入力してください。]** | 署名者の証明書のロケーションとファイル名を指定します。<br><br>この情報を指定した場合、生成された証明書は署名済み証明書です。この情報を指定しなかった場合、生成された証明書は自己署名ルート証明書です。<br><br>このオプションは、-s オプションが指定されている場合にのみ適用されます。 |
| **-ca** *y* \| *n* | **[認証局 (はい(y) またはいいえ (n))]** | 他の証明書に署名できる新消極証明書を作成します。デフォルトでは、証明書は認証局ではありません (n)。<br><br>このオプションは、-s オプションが指定されている場合にのみ適用されます。 |
| **-ck** *filename* | **[署名者のプライベートキーのファイルパスを入力してください。]** | 証明書要求に関連付けられたプライベートキーを保存するロケーションとファイル名を指定します。このオプションは、-c オプションにファイル名を指定した場合にのみ表示されます。<br><br>このオプションは、-s オプションが指定され、-c オプションにファイル名を指定した場合にのみ適用されます。 |

| オプション | プロンプト名 | 説明 |
| --- | --- | --- |
| **-co** *filename* | **[証明書を保存するファイル名を入力してください。]** | 指定されたファイルにパブリック証明書を書き込みます。<br><br>証明書を保存するロケーションとファイル名を指定します。ロケーションとファイル名を指定しなければ、証明書は保存されません。<br><br>このオプションは、-s オプションが指定されている場合にのみ適用されます。 |
| **-cp** *password* | **[署名者のプライベートキーのパスワードを入力してください。]** | 署名者のプライベートキーの暗号化に使用されたパスワードを指定します。<br><br>このパスワードは、プライベートキーが暗号化されている場合にのみ指定します。<br><br>このオプションは、-s オプションが指定されている場合にのみ適用されます。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | プロンプト名 | 説明 |
|---|---|---|
| **-ec** *curve* | **[ECC 曲線]** | このオプションは、ECC 暗号化のライセンスを購入し、かつ ECC で -t オプションを選択した場合にかぎり表示されます。デフォルトは secp521r1 です。<br><br>このオプションは、-r オプションが指定されている場合にのみ適用されます。<br><br>次の ECC 曲線がサポートされています。<br><br>● sect163k1<br>● sect163r2<br>● sect233k1<br>● sect233r1<br>● sect283k1<br>● sect283r1<br>● sect409k1<br>● sect409r1<br>● sect571k1<br>● sect571r1<br>● secp192r1<br>● secp224r1<br>● secp256r1<br>● secp384r1<br>● secp521r1<br><br>曲線名の 3 桁の数値は、曲線のビットサイズを表します。曲線の暗号化強度は、半分のサイズの AES キーとほぼ同等です。たとえば、512 ビットの ECC 曲線は、256 ビットの AES キーと同じ強度になります。 |
| **-io** *filename* | **[ID を保存するファイル名を入力してください。]** | 生成された ID を保存するロケーションとファイル名を指定します。<br><br>このオプションは、-s オプションが指定されている場合にのみ適用されます。 |

| オプション | プロンプト名 | 説明 |
| --- | --- | --- |
| **-ko** *filename* | **[プライベートキーを保存するファイル名を入力してください。]** | プライベートキーを保存するロケーションとファイル名を指定します。 |
| **-kp** *password* | **[プライベートキーを保護するためのパスワードを入力してください。]** | プライベートキーを暗号化するために使用するパスワードを指定します。<br><br>パスワードを指定しない場合、プライベートキーは暗号化されません。このオプションは、-ko オプションを使用してファイル名を指定した場合にのみ適用されます。 |
| **-m** *value* | **[シリアル番号]** | シリアル番号を指定します。シリアル番号は 40 桁以下の 16 進数文字列である必要があります。この番号は、現在の署名者が署名したすべての証明書でユニークである必要があります。シリアル番号を指定しなかった場合、シリアル番号として GUID が生成されます。<br><br>このオプションは、-s オプションが指定されている場合にのみ適用されます。 |
| **-r** | N/A | PKCS10 証明書を生成します。<br><br>このオプションを指定すると、次のオプションは適用されません。<br><br>● -c<br>● -ca<br>● -ck<br>● -co<br>● -cp<br>● -m<br>● -io<br>● -u<br>● -x<br>● -v |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | プロンプト名 | 説明 |
|---|---|---|
| **-ro** *filename* | **[要求を保存するファイル名を入力してください。]** | PCKS10 証明書を保存するロケーションとファイル名を指定します。<br><br>このオプションは、-r オプションを指定した場合にかぎり適用されます。 |
| **-s** *filename* | N/A | 署名された PKCS10 証明書要求のロケーションと名前を指定します。証明書要求は、DER または PEM でコード化できます。このオプションを指定すると、次のオプションは適用されません。<br><br>● -b<br>● -ec<br>● -sc<br>● -scn<br>● -sl<br>● -so<br>● -sou<br>● -sst<br>● -t<br>● -x |
| **-sc** *value* | サブジェクト情報 | サブジェクト国コードを指定します。<br><br>このオプションは、-r オプションが指定されている場合にのみ適用されます。 |
| **-scn** *value* | サブジェクト情報 | サブジェクト通称を指定します。<br><br>このオプションは、-r オプションが指定されている場合にのみ適用されます。 |
| **-sl** *value* | サブジェクト情報 | サブジェクト地方を指定します。<br><br>このオプションは、-r オプションが指定されている場合にのみ適用されます。 |

| オプション | プロンプト名 | 説明 |
| --- | --- | --- |
| **-so** *value* | サブジェクト情報 | サブジェクト組織を指定します。<br><br>このオプションは、-r オプションが指定されている場合にのみ適用されます。 |
| **-sou** *value* | サブジェクト情報 | サブジェクト組織単位を指定します。<br><br>このオプションは、-r オプションが指定されている場合にのみ適用されます。 |
| **-sst** *value* | サブジェクト情報 | サブジェクト都道府県を指定します。<br><br>このオプションは、-r オプションが指定されている場合にのみ適用されます。 |
| **-t** *type* | **[**暗号化タイプを選択してください。**]** | 暗号化タイプを指定します。ECC 暗号化のライセンスを購入している場合は、暗号化タイプを指定します。RSA または ECC を指定します。<br><br>このオプションは、-r オプションが指定されている場合にのみ適用されます。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | プロンプト名 | 説明 |
| --- | --- | --- |
| **-u** *value,...* | **[キーの使用法]** | 証明書の使用方法を指定します。<br><br>証明書のプライベートキーをどのように使用できるかを示す番号をカンマ区切りのリストで指定します。これは詳細オプションであり、ほとんどの状況ではデフォルト設定で十分です。デフォルトは、証明書が認証局であるかどうかによって異なります。証明書認証局のある証明書のデフォルトは、**6,7** です。証明書認証局のない証明書のデフォルトは、**3,4,5** です。<br><br>このオプションは、-s オプションが指定されている場合にのみ適用されます。<br><br>次のような値があります。<br><br>● 1. デジタル署名<br>● 2. 否認防止<br>● 3. キーの暗号化<br>● 4. データの暗号化<br>● 5. キーの承諾<br>● 6. 証明書の署名<br>● 7. CRL の署名<br>● 8. 暗号化のみ<br>● 9. 復号化のみ |
| **-v** *years* | **[証明書の有効期間 (年数) (1-100)]** | 証明書が有効である年数 (1 ～ 100) を指定します。この期間を過ぎると、証明書と、その証明書で署名されたすべての証明書の有効期限が切れます。<br><br>このオプションは、-s オプションが指定されている場合にのみ適用されます。 |
| **-x** | N/A | 自己署名証明書を生成します。<br><br>このオプションは、-s オプションが指定されている場合にのみ適用されます。 |

### 備考

ユーザーは、一般的にサードパーティから証明書を購入します。そのような認証局には、証明書を作成するために独自のツールが用意されています。次のツールは、開発用やテスト用の証明書を作成するときに特に便利ですが、稼働時の証明書に使用することもできます。

オプションを指定せず、createcert から必要な値を要求を受けることで、署名証明書を作成することができます。または、証明書に必要なオプションを指定することもできます。

あるユーザーが要求を作成して別のユーザーがそれに署名するというように、処理を 2 つのステップに分ける場合、1 番目のユーザーは -r を指定した createcert を実行して要求を作成し、2 番目のユーザーは -s を指定した createcert を実行して要求に署名することができます。

エンタープライズルート証明書 (他の証明書に署名する証明書) を作成するためには、認証局を使用して自己署名ルート証明書を作成し、デフォルトの署名を使用することをおすすめします。たとえば、**-ca** オプションに **y** を指定し、**-u** オプションに **6,7** を指定します。

エンタープライズルート証明書 (他の証明書に署名する証明書) を生成するためには、認証局を使用して自己署名ルート証明書を作成し、デフォルトの署名を使用することをおすすめします。

### 参照


- 「証明書」1410 ページ
- 「証明書ビューアーユーティリティ (viewcert)」934 ページ
- 「-ec dbeng12/dbsrv12 サーバーオプション」223 ページ
- 「Encryption (ENC) 接続パラメーター」342 ページ
- 「FIPS 認定の暗号化テクノロジ」1412 ページ


### 例

次の例では、次の特性を使用して自己署名ルート証明書を作成します。

- 1024 ビット RSA キー
- 生成されたシリアル番号
- 5 年間有効
- 証明書認証局ではない
- キーの暗号化、データの暗号化、キーの承諾で使用される
- パスワードが sql のプライベートキー

```
createcert.exe -t rsa -b 1024 -sc CA -sst ON -sl Waterloo -so MyOrganizationalUnit -sou MyCompany -
scn test -x -m 0 -v 5 -ca 0 -u 3,4,5 -co root-cert.pem -ko root-key.pem -io root-id.pem -kp sql
```

次の例では、次の特性を使用して自己署名ルート証明書を作成します。

- ECC sect283ra キー
- 生成されたシリアル番号
- 2 年間有効

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

●証明書認証局 (他の証明書に署名可能)

●キーの暗号化、データの暗号化、キーの承諾、証明書の署名で使用される

●パスワードが sql のプライベートキー

```
createcert.exe -t ecc -ec sect283ra -sc CA -sst ON -sl Waterloo -so MyOrganizationalUnit -sou
MyCompany -scn test -x -m 0 -v 2 -ca 1 -u 3,4,5,6 -co root-cert.pem -ko root-key.pem -io root-id.pem -
kp sql
```

次の例では、次の特性を使用して ca キーで署名された証明書を作成します。

●1024 ビット RSA キー

●生成されたシリアル番号

●5 年間有効

●証明書認証局ではない

●キーの暗号化、データの暗号化、キーの承諾で使用される

●パスワードが sql のプライベートキー

```
createcert.exe -t rsa -b 1024 -sc CA -sst ON -sl Waterloo -so MyOrganizationalUnit -sou MyCompany -
scn test -c ca-cert.pem -ck ca-key.pem -cp cakeypass -m 0 -v 5 -ca 0 -u 3,4,5 -co root-cert.pem -ko root-
key.pem -io root-id.pem -kp sql
```

次の例では、署名済み証明書を作成します。この例では、署名者の証明書にファイル名を指定しないため、自己署名ルート証明書になります。

```
createcert
SQL Anywhere X.509 証明書ジェネレーター バージョン 12.0.1.3330
暗号化タイプを選択してください ((R)SA または (E)CC) : r
RSA のキー長を入力してください (512 ～ 16384) : 1024
キーペア作成中...
国コード : CA
都道府県 : Ontario
地方 : Waterloo
組織 : Sybase iAnywhere
組織単位 : Engineering
通称 : Test Certificate
署名者の証明書のファイルパスを入力してください :
証明書は自己署名ルート
シリアル番号 [生成された GUID] :
生成されたシリアル番号 : bfb89a26fb854955954cabc4d056e177
証明書の有効期間 (年数) (1-100) : 10
証明書認証局 (Y/N) [N] : n
1. デジタル署名
2. 否認防止
3. キーの暗号化
4. データの暗号化
5. キーの承諾
6. 証明書の署名
7. CRL の署名
8. 暗号化のみ
9. 復号化のみ
キーの使用法 [3、4、5] : 3,4,5
証明書を保存するファイル名を入力してください : cert.pem
```

```
プライベートキーを保存するファイル名を入力してください : key.pem
プライベートキーを保護するためのパスワードを入力してください : pwd
ID を保存するファイル名を入力してください : id.pem
```

次の例では、デフォルトの署名を使用し、認証局を使用する自己署名ルート証明書である、エンタープライズルート証明書 (他の証明書に署名する証明書) を作成します。**[証明書認証局]** プロンプトへの応答または -ca オプションが **y** で、**[キーの使用法]** プロンプトへの応答または -u オプションが **6,7** (デフォルト) となります。

```
証明書認証局 (Y/N) [N] : y
1. デジタル署名
2. 否認防止
3. キーの暗号化
4. データの暗号化
5. キーの承諾
6. 証明書の署名
7. CRL の署名
8. 暗号化のみ
9. 復号化のみ
キーの使用法 [6、7] : 6,7
```

# 証明書ビューアーユーティリティ (viewcert)

パブリックキーインフラストラクチャー (PKI) オブジェクト内の値の表示、PKI オブジェクトのエンコーディングの変換、プライベートキーの暗号化と復号化を行います。

### 構文

**viewcert** [ *options* ] *input-file*

| オプション | 説明 |
|---|---|
| **-d** | 出力を DER でコード化します。このオプションは、-o オプションと一緒に指定した場合のみ役立ちます。-p とは一緒に使用できません。デフォルトでは、判読可能なテキスト形式で PKI オブジェクトが出力されます。 |
| **-ip** *input-password* | *input-file* に暗号化されたプライベートキーが含まれる場合、プライベートキーの復号化に必要なパスワードを指定します。 |
| **-o** *output-file* | 出力が書き込まれるファイルを指定します。デフォルトでは、viewcert を実行しているコマンドプロンプトウィンドウに出力が書き込まれます。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| **-op** *output-password* | プライベートキーを暗号化するために使用されるパスワードを指定します。このオプションは、-d または -p と一緒に指定した場合のみ役立ちます。デフォルトでは、プライベートキーは暗号化されません。 |
| **-p** | 出力を PEM でコード化します。このオプションは、-o オプションと一緒に指定した場合のみ役立ちます。-d とは一緒に使用できません。デフォルトでは、判読可能なテキスト形式で PKI オブジェクトが出力されます。 |
| *input-file* | DER または PEM でコード化された PKI オブジェクトである必要があるファイルを指定します。 |

## 備考

viewcert ユーティリティを使用して表示できる PKI オブジェクトのタイプは、次のとおりです。

- X.509 証明書
- 証明書要求
- プライベートキー
- 証明書失効リスト (CRL)

viewcert は、DER と PEM の間でコード化タイプを変換したり、プライベートキーを暗号化または復号化したりするために使用することもできます。

viewcert ユーティリティは、RSA オブジェクトと ECC オブジェクトをサポートします。ECC オブジェクトを表示するには、別途ライセンスを注文する必要があります。「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

## 参照

- 「証明書作成ユーティリティ (createcert)」924 ページ

## 例

次の例では、SQL Anywhere に付属するサンプル RSA 証明書を表示できます。

```
viewcert rsaroot.crt
```

この例は、次の出力を生成します。

```
SQL Anywhere X.509 証明書ビューアー バージョン 12.0.1.3149

X.509 証明書
-----------------
通称:          RSA Root
国コード:        test
都道府県:        test
```

```
地方:          test
組織:          test
組織単位:       test
発行元:         RSA Root
シリアル番号:    303031
発行済み:       Apr 15, 2002  09:53:51
執行:          Apr 16, 2022  09:53:51
シグネチャーアルゴリズム:    RSA, MD5
キータイプ:      RSA
キーサイズ:      1024 bits
基本的な制約:    認証局であり、パス長制限は次のとおり: 10
キーの使用法:    証明書の署名、CRL の署名
```

# データソースユーティリティ (dbdsn)

SQL Anywhere ODBC データソースの作成、削除、説明、およびリスト作成を行います。

### 構文

**dbdsn** [ *modifier-options* ]
{ **-l**[ **s** | **u** ]
| **-d**[ **s** | **u** ] *dsn*
| **-g**[ **s** | **u** ] *dsn*
| **-w**[ **s** | **u** ] *dsn* [*details-options*;...]
| **-cl** }

| 主要オプション | 説明 |
|---|---|
| *@data* | 指定された環境変数または設定ファイルからオプションを読み込みます。「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル非表示ユーティリティを使用して、設定ファイルの内容を読みにくくすることができます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |
| **-l**[ **s** \| **u** ] | 使用可能な SQL Anywhere ODBC データソースをリストします。リストの形式は -b オプションまたは -v オプションを使用して変更できます。Windows では、**u** (ユーザー) または **s** (システム) 指定子を使用して、オプションを変更できます。デフォルトの指定子は **u** です。 |
| **-d**[ **s** \| **u** ] *dsn* | 指定の SQL Anywhere データソースを削除します。-y を指定すると、確認メッセージを表示せずに既存のデータソースが削除されます。Windows では、**u** (ユーザー) または **s** (システム) 指定子を使用して、オプションを変更できます。デフォルトの指定子は **u** です。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 主要オプション | 説明 |
| --- | --- |
| **-g**[ **s** \| **u** ] *dsn* | 指定された SQL Anywhere データソースの定義をリストします。出力のフォーマットは、-b オプションまたは -v オプションを使用して変更できます。Windows では、**u** (ユーザー) または **s** (システム) 指定子を使用して、オプションを変更できます。デフォルトの指定子は **u** です。 |
| **-w**[ **s** \| **u** ] *dsn* [ *details-options* ] | 新しいデータソースを作成します。同じ名前のデータソースが存在する場合は上書きします。-w オプションと一緒に -c オプションを指定する必要があります。-y を指定すると、確認メッセージを表示せずに既存のデータソースを上書きします。Windows では、**u** (ユーザー) または **s** (システム) 指定子を使用して、オプションを変更できます。デフォルトの指定子は **u** です。 |
| **-cl** | dbdsn ユーティリティがサポートしている接続パラメーターをリストします。「接続パラメーター」320 ページを参照してください。<br><br>サポートされている ODBC 接続パラメーターの詳細については、「ODBC 接続パラメーター」942 ページを参照してください。 |

| 変更オプション | 説明 |
| --- | --- |
| **-b** | リストの出力を 1 行の接続文字列にフォーマットします。 |
| **-cm** | データソースの作成に使用するコマンドを表示します。このオプションを使用すると、作成コマンドをファイルに出力できます。作成コマンドは別のコンピューターにデータソースを追加したり、変更が加えられたデータソースを元の状態にリストアするために使用できます。-cm とともに -g オプションまたは -l オプションを指定しないとコマンドは失敗します。-g を指定すると、指定のデータソースの作成コマンドが表示されるのに対し、-l を指定するとすべてのデータソースの作成コマンドが表示されます。<br><br>指定のデータソースが存在しない場合は、データソースを削除するコマンドが生成されます。たとえば、コンピューターに mydsn データソースが存在しない場合、dbdsn -cm -g mydsn は mydsn データソースを削除する次のコマンドを返します。<br><br>dbdsn -y -du "mydsn" |

| 変更オプション | 説明 |
| --- | --- |
| **-dr** | データソースを表示するときに Driver パラメーターを含めます。これにより、-cm オプションを使用してデータソースを再作成するときに、dbdsn の現バージョンで異なるバージョンの ODBC ドライバーを参照するデータソースを作成することができます。<br><br>たとえば、次のようなコマンドを使用してバージョン 11 のデータソースを作成します。<br><br>dbdsn -y -wu "11.0 Student Sample" -c "UID=DBA;PWD=sql;...;Driver=SQL Anywhere 11"<br><br>dbdsn -cm -l を実行すると、同じコマンドが Driver= パラメーターなしで表示されます。これによって、SQL Anywhere バージョン 12 の ODBC ドライバーを使用して ODBC データソースが再作成されます。<br><br>それに対し、dbdsn -dr -cm -l を実行すると、コマンドに「Driver=parameter」が含まれるため、バージョン 11 の ODBC ドライバーを使用した場合とまったく同じデータソースを再作成できます。 |
| **-f** | 使用されているシステムファイルの名前を表示します。このオプションは UNIX でのみ使用できます。 |
| **-ns** | 環境変数の設定を使用してシステム情報ファイル (デフォルトのファイル名は *.odbc.ini*) のロケーションを特定するように指定します。このオプションは、使用される可能性のあるシステム情報ファイルが複数ある場合に、dbdsn が使用するファイルを特定する目的にも利用できます。このオプションは UNIX でのみ使用できます。<br><br>データソースを作成するときに -ns オプションを指定しなかった場合、システム情報ファイルはユーザーのホームフォルダーとそのパス内で検索されます。<br><br>システム情報ファイルがどのように検索されるかについては、「UNIX での ODBC データソース」48 ページを参照してください。 |
| **-o** *filename* | 指定したファイルに、出力メッセージを書き込みます。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 変更オプション | 説明 |
| --- | --- |
| **-or** | -c オプションとともに指定すると、iAnywhere Solutions Oracle ドライバーのデータソースが作成されます。次に例を示します。<br><br>dbdsn -w MyOracleDSN -or -c Userid=DBA;Password=sql;ServiceName=abcd;ArraySize=500;ProcResults=y<br><br>-cl オプションを -or オプションとともに指定すると、iAnywhere Solutions Oracle ドライバーの接続パラメーターのリストを取得できます。<br><br>「iAnywhere Solutions 12 - Oracle ODBC ドライバー」『Mobile Link サーバー管理』を参照してください。 |
| **-pe** | Password 接続パラメーターで指定されたパスワードを暗号化し、ENP 接続パラメーターを使用して、暗号化されたパスワードをデータソースに格納します。 |
| **-q** | データベースサーバーメッセージウィンドウに出力を表示しないようにします。データソースの削除時または変更時に -q を指定する場合は、-y も指定してください。 |
| **-v** | 数行のリスト出力を表としてフォーマットします。 |
| **-y** | 確認を要求するプロンプトを表示しないで、各データソースを削除または上書きします。データソースの削除時または変更時に -q を指定する場合は、-y も指定してください。 |

| 詳細オプション | 説明 |
| --- | --- |
| **-c "***keyword=value;...***"** | 接続パラメーターを接続文字列として指定します。「接続パラメーター」320 ページを参照してください。 |
| **-cw** | (-c で指定された) DBF パラメーターが確実に絶対ファイル名になるようにします。DBF の値が絶対ファイル名でない場合、dbdsn は現在の作業ディレクトリ (CWD) を付加します。オペレーティングシステムの中には、バッチファイルですぐに利用できる CWD 情報がないものもあるので、このオプションは便利です。 |

**備考**

変更オプションは、主要オプションの前または後に指定できます。

データソースユーティリティは、プラットフォームを問わないユーティリティで、ODBC データソースアドミニストレーターに代わり SQL Anywhere ODBC データソースの作成、削除、記述、リストを実行します。このユーティリティは、バッチ処理に役立ちます。

**警告**
ユーザー ID、パスワード (暗号化の有無は不問)、データベースキーをデータソースに保存するのは安全な方法ではありません。データベースに機密データが含まれている場合は、このような情報をデータソースに保存しないでください。

Windows オペレーティングシステムでは、データソースはレジストリに保存されます。

ODBC データソースアドミニストレーターを使用した Windows でのデータソース作成の詳細については、「ODBC データソース」36 ページを参照してください。

UNIX オペレーティングシステムでは、データソースはシステム情報ファイル (デフォルトのファイル名は *.odbc.ini*) に保持されます。データソースユーティリティを使用して SQL Anywhere ODBC データソースを UNIX 上で作成または削除すると、システム情報ファイルの [ODBC Data Sources] セクションが自動的に更新されます。UNIX で -c オプションを使用して Driver 接続パラメーターを指定しない場合、データソースユーティリティによって SQLANY12 環境変数の設定に基づき SQL Anywhere ODBC ドライバーのフルパスを使って Driver エントリが自動的に追加されます。

システム情報ファイルの詳細については、「UNIX での ODBC データソース」48 ページを参照してください。

サポートされている ODBC 接続パラメーターの詳細については、「ODBC 接続パラメーター」942 ページを参照してください。

**警告**
SQL Anywhere データソースのみを使用する場合以外は、UNIX でファイル難読化ユーティリティ (dbfhide) を使って *.odbc.ini* システム情報ファイルを難読化しないでください。他のデータソース (Mobile Link 同期など) を使用する予定の場合、システム情報ファイルを難読化すると、他のドライバーが正しく機能しなくなることがあります。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』を参照してください。

## 参照


- 「ODBC データソース」36 ページ
- 「UNIX での ODBC データソース」48 ページ


## 例

データソース newdsn の定義を書き込みます。データソースがすでに存在する場合でも、確認メッセージは表示しません。

```
dbdsn -y -w newdsn -c "UID=DBA;PWD=sql;Host=myhost"
```

次のように、オプションを別の順序で指定することもできます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
dbdsn -w newdsn -c "UID=DBA;PWD=sql;Host=myhost" -y
```

既知のすべてのユーザーデータソースをリストします (1 行に 1 つのデータソース名)。

```
dbdsn -l
```

既知のすべてのシステムデータソースをリストします (1 行に 1 つのデータソース名)。

```
dbdsn -ls
```

すべてのデータソースを関連する接続文字列とともにリストします。

```
dbdsn -l -b
```

ユーザーデータソース MyDSN 用の接続文字列をレポートします。

```
dbdsn -g MyDSN
```

システムデータソース MyDSN 用の接続文字列をレポートします。

```
dbdsn -gs MyDSN
```

最初に BadDSN の接続パラメーターをリストし、確認メッセージを表示してから、データソース BadDSN を削除します。

```
dbdsn -d BadDSN -v
```

確認メッセージを表示せずに、データソース BadDSN を削除します。

```
dbdsn -d BadDSN -y
```

データベースサーバー MyServer のデータソース NewDSN を作成します。

```
dbdsn -w NewDSN -c "UID=DBA;PWD=sql;Server=MyServer"
```

NewDSN がすでに存在する場合は、データソースを上書きするかどうか確認を求められます。

すべての接続パラメーター名とそのエイリアスをリストします。

```
dbdsn -cl
```

すべてのユーザーデータソースをリストします。

```
dbdsn -l -o dsninfo.txt
```

すべての接続パラメーター名をリストします。

```
dbdsn -cl -o dsninfo.txt
```

絶対ファイル名を指定します。ODBC データソースは、*DBF=c:¥SQLAnywhere12¥my.db* という名前で作成されます。

```
c:¥SQLAnywhere12> dbdsn -w testdsn -cw -c UID=DBA;PWD=sql;Server=SQLAny;DBF=my.db
```

次のコマンドを実行して SQL Anywhere 12 Demo データソースを作成し、それを *restoredsn.bat* というファイルに出力します。

```
dbdsn -cm -gs "SQL Anywhere 12 Demo" > restoredsn.bat
```

*restoredsn.bat* ファイルには以下が含まれます。

```
dbdsn -y -ws "SQL Anywhere 12 Demo" -c "UID=DBA;PWD=sql;
DBF='C:¥Documents and Settings¥All Users¥Documents¥SQL Anywhere 12¥Samples¥demo.db';
Server=demo12;START='C:¥Program Files¥SQL Anywhere 12¥Bin32¥dbeng12.exe';
ASTOP=yes;Description='SQL Anywhere 12 Sample Database'"
```

次のコマンドは、UNIX のシステム情報ファイルのロケーションを返します。

```
dbdsn -f
```

このコマンドは次の出力を返します。

```
dbdsn using /home/user/.odbc.ini
```

システム情報ファイルのロケーションを変更します。

```
export ODBCINI=./myodbc.ini
```

dbdsn -f を使用してシステム情報ファイルの新しいロケーションを確認します。

```
dbdsn using ./myodbc.ini
```

-ns オプションを使用してデータソースを作成します。

```
dbdsn -w NewDSN -c "UID=DBA" -ns
```

このコマンドは次の出力を返します。

```
Configuration "newdsn" written to file ./myodbc.ini
```

## ODBC 接続パラメーター

データソースユーティリティ (dbdsn) は以下の ODBC 接続パラメーターをサポートしています。ブール (true または false) 引数は、true の場合は YES または 1、false の場合は NO または 0 のいずれかです。

| 名前 | 説明 |
| --- | --- |
| Delphi | Delphi では、1 ローにつき複数のブックマーク値を処理できません。この値を NO に設定すると、各ローに 1 つ (YES の場合は 2 つ) のブックマーク値が割り当てられます。このオプションを YES に設定すると、スクロール可能なカーソルのパフォーマンスが向上します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 名前 | 説明 |
| --- | --- |
| DescribeCursor | このパラメーターにより、プロシージャーが呼び出されたときにカーソルを再記述する頻度を指定できます。デフォルト設定は [任意] です。<br><br>● **[しない]**　カーソルを再記述する必要がない場合には、このオプションを 0、N、または NO に指定します。カーソルの再記述は負荷が高く、パフォーマンスを低下させる可能性があります。<br><br>● **[任意]**　カーソルを再記述する必要があるかどうかを ODBC ドライバーが決定するようにするには、1、Y、または YES を指定します。プロシージャーに RESULT 句があると、ODBC アプリケーションは、カーソルを開いた後結果セットを再記述できません。これがデフォルト設定です。<br><br>● **[常に]**　2、A、または ALWAYS を指定すると、カーソルを開くたびに再記述します。Transact-SQL プロシージャーや、複数の結果セットを返すプロシージャーを使用する場合は、カーソルを開くたびに再記述する必要があります。 |
| Description | このパラメーターにより、ODBC データソースの説明を入力できます。 |

| 名前 | 説明 |
| --- | --- |
| Driver | このパラメーターでは、接続の ODBC ドライバーを Driver=driver-name の形式で指定できます。デフォルトで使用されるドライバーは **SQL Anywhere 12** です。*driver-name* は、SQL Anywhere *X* にしてください。その場合、*X* にはソフトウェアのメジャーバージョン番号を指定します。*driver-name* が **SQL Anywhere** で始まらない場合、データソースユーティリティ (dbdsn) によって読み取ることができません。<br><br>UNIX では、このパラメーターによって共有オブジェクトへの完全に修飾されたパスが指定されます。UNIX で Driver 接続パラメーターを指定しない場合、データソースユーティリティによって SQLANY12 環境変数の設定に基づき SQL Anywhere ODBC ドライバーのフルパスを使って Driver エントリが自動的に追加されます。 |
| Escape | このパラメーターは、テーブルまたはカラムのリストを返すときに ODBC ドライバーで生成される、SQL 文の LIKE 句で使用されるエスケープ文字を指定します。<br><br>デフォルトでは、ODBC ドライバーでチルダ文字 (~) が使用されますが、一部のアプリケーションではエスケープ文字に円記号 (¥) が使用されることもあります。<br><br>次の接続文字列フラグメントは、エスケープ文字に円記号 (¥) を指定します。<br><br>"DSN=SQL Anywhere 12 Demo;UID=DBA;PWD=sql;ESCAPE=¥¥;Host=myhost" |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 名前 | 説明 |
| --- | --- |
| GetTypeInfoChar | このオプションを YES に設定すると、CHAR カラムは SQL_VARCHAR ではなく SQL_CHAR として返されます。デフォルトでは、CHAR カラムは SQL_VARCHAR として返されます。 |
| InitString | InitString により、接続確立後すぐに実行されるコマンドを指定できます。たとえば、データベースオプションを設定したり、ストアドプロシージャーを呼び出したりできます。 |

| 名前 | 説明 |
| --- | --- |
| IsolationLevel | 以下の値の 1 つを指定して、データソースの初期独立性レベルを設定できます。<br><br>● **0**　これはコミットされない読み出し独立性レベルとも呼ばれます。これはデフォルトの独立性レベルです。これは最大レベルの同時実行性を提供しますが、結果セットにダーティリード、繰り返し不可能読み出し、幻ローが発生する場合があります。<br><br>● **1**　これはコミットされた読み出しレベルとも呼ばれます。レベル 0 よりも低い同時実行性を提供しますが、レベル 0 の結果セットに見られる不整合性が一部解消されます。繰り返し不可能読み出しや幻ローが発生することはありますが、ダーティリードは発生しません。<br><br>● **2**　これは繰り返し可能読み出しレベルとも呼ばれます。幻ローが発生することがあります。ダーティリードと繰り返し不可能ローは発生しません。<br><br>● **3**　これは、直列化可能レベルとも呼ばれます。これは最低レベルの同時実行性を提供する、最も厳しい独立性レベルです。ダーティリード、繰り返し不可能読み出し、幻ローは発生しません。<br><br>● **Snapshot**　この独立性レベルを使用するには、データベースのスナップショットアイソレーションを有効にする必要があります。スナップショットアイソレーションのレベルは、読み込みと書き込み間の干渉を防ぎます。書き込みは相互に干渉する可能性があります。競合の場合、一貫性のない動作が多少生じる可能性がありますが、パフォーマンスは独立性レベ |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 名前 | 説明 |
| --- | --- |
| | ルを 0 に設定した場合と同じです。<br><br>● **Statement-snapshot**　この独立性レベルを使用するには、データベースのスナップショットアイソレーションを有効にする必要があります。スナップショットアイソレーションのレベルは、読み込みと書き込み間の干渉を防ぎます。書き込みは相互に干渉する可能性があります。競合の場合、一貫性のない動作が多少生じる可能性がありますが、パフォーマンスは独立性レベルを 0 に設定した場合と同じです。<br><br>● **Readonly-statement-snapshot**　この独立性レベルを使用するには、データベースのスナップショットアイソレーションを有効にする必要があります。スナップショットアイソレーションのレベルは、読み込みと書き込み間の干渉を防ぎます。書き込みは相互に干渉する可能性があります。競合の場合、一貫性のない動作が多少生じる可能性がありますが、パフォーマンスは独立性レベルを 0 に設定した場合と同じです。<br><br>「独立性レベル選択のガイドライン」『SQL Anywhere サーバー SQL の使用法』を参照してください。 |

| 名前 | 説明 |
| --- | --- |
| KeysInSQLStatistics | SQLStatistics 関数によって外部キーが返されるようにするには、このパラメーターを YES に設定します。ODBC 仕様では、SQLStatistics によってプライマリキーと外部キーが戻されないように指定しています。ただし、一部の Microsoft アプリケーション (Microsoft Visual Basic や Microsoft Access など) では、SQLStatistics によってプライマリキーと外部キーが戻されることを前提にしています。 |
| LazyAutocommit | 文が完了するまでコミット操作を遅延させるには、このパラメーターを YES に設定します。 |
| PrefetchOnOpen | PrefetchOnOpen が YES に設定されていると、カーソルを開く要求を含むプリフェッチ要求が送信されます。プリフェッチにより、カーソルを開くたびにネットワーク要求が行われることはなくなります。カーソルを開くときにプリフェッチを実行するには、カラムをバインドしておきます。この接続パラメーターにより、クライアント／サーバー要求の数が削減され、LAN や WAN を介したパフォーマンスが向上します。 |
| PreventNotCapable | SQL Anywhere ODBC ドライバーは、修飾子をサポートしていないためエラーを返します。ODBC アプリケーションの中には、このエラーを適切に処理しないものもあります。このようなアプリケーションでも作業できるように、このエラーコードが返されないようにするには、このパラメーターを YES に設定します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 名前 | 説明 |
| --- | --- |
| SuppressWarnings | フェッチ時にデータベースサーバーから返される警告メッセージを表示しない場合は、このパラメーターを YES に設定します。バージョン 8.0.0 以降のデータベースサーバーでは、それよりも前のバージョンのソフトウェアに比べて多様なフェッチ警告が返されます。以前のバージョンのソフトウェアを使用して配備されたアプリケーションに対して、フェッチの警告を適切に処理するためにこのオプションを選択できます。 |
| TranslationDLL | このオプションは下位互換性のために提供されています。トランスレータの使用はおすすめできません。 |
| TranslationName | このオプションは下位互換性のために提供されています。トランスレータの使用はおすすめできません。 |
| TranslationOption | このオプションは下位互換性のために提供されています。トランスレータの使用はおすすめできません。 |

# dbisqlc ユーティリティ (旧式)

dbisqlc ユーティリティは、データベースで SQL 文を実行します。

**注意**
dbisqlc ユーティリティでは、SQL でサポートされるすべての SQL 文と機能がサポートされているわけではなく、現バージョンのデータベースサーバーで使用できるすべての機能がサポートされない場合もあります。Interactive SQL ユーティリティの使用をおすすめします。「Interactive SQL ユーティリティ (dbisql)」975 ページを参照してください。

### 構文

**dbisqlc** [ *options* ] [ *dbisqlc-statement* | *dbisql-script-file* ]

| オプション | 説明 |
| --- | --- |
| **-c** *"keyword=value; ..."* | 接続パラメーターを指定します。Interactive SQL が接続できない場合は、接続パラメーターを入力するウィンドウが表示されます。「接続パラメーター」320 ページを参照してください。 |
| **-d** *delimiter* | コマンドデリミターを指定します。デリミターを囲む引用符は省略可能ですが、コマンドシェル自体がデリミターを特別な方法で解釈するときは必ず指定します。<br><br>指定したコマンドデリミターは、現在の dbisqlc セッションのすべての接続で使用できます。 |
| **-q** | 出力メッセージを表示しません。これは、文またはスクリプトファイルを使って Interactive SQL を起動したときにのみ便利です。このオプションを指定した場合、エラーメッセージは表示されますが、次の情報は表示されません。<br><br>●警告およびその他の致命的でないメッセージ<br><br>●結果セットの出力 |
| **-x** | 文をスキャンしますが、実行しません。長いスクリプトファイルの構文エラーをチェックする場合に有用です。 |

## 備考

dbisqlc ユーティリティを使用すると、SQL 文を入力したり、Interactive SQL スクリプトファイルを実行したりすることができます。

*dbisqlc-statement* を指定すると、dbisqlc がその文を実行します。スクリプトファイル名も指定できます。*dbisqlc-statement* または *dbisql-script-file* 引数を指定しなかった場合、dbisqlc は対話型モードになります。このモードでは、文をコマンドウィンドウに入力できます。

dbisqlc ユーティリティは、Microsoft Windows、Mac OS X、UNIX で使用できます。

## 参照


- 「Interactive SQL ユーティリティ (dbisql)」975 ページ
- 「SQL 言語の要素」『SQL Anywhere サーバー SQL リファレンス』


 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 例

次のコマンドを実行すると、ユーザー ID DBA とパスワード sql で、現在のデフォルトサーバーに対してスクリプトファイル *mycom.sql* が実行されます。スクリプトファイルにエラーがあった場合は、処理は停止します。

```
dbisqlc -c "UID=DBA;PWD=sql" mycom.sql
```

次のコマンドを実行すると、現在のデフォルトデータベースにユーザーが追加されます。

```
dbisqlc -c "UID=DBA;PWD=sql" CREATE USER joe IDENTIFIED BY passwd
```

## 消去ユーティリティ (dberase)

データベースに関連する DB 領域とトランザクションログファイルを消去します。

### 構文

**dberase** [ *options* ] *database-file*

| オプション | 説明 |
|---|---|
| **@***data* | 指定された環境変数または設定ファイルからオプションを読み込みます。「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル非表示ユーティリティを使用して、設定ファイルの内容を読みにくくすることができます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |
| **-ek** *key* | 強力に暗号化されているデータベースの暗号化キーをコマンドに直接指定します。データベースが強力に暗号化されている場合、データベースまたはトランザクションログを使用するには必ず暗号化キーを指定します。強力な暗号化が適用されたデータベースの場合、-ek または -ep のどちらかを指定します。両方同時には指定できません。強力に暗号化されたデータベースでは、正しいキーを指定しないとコマンドが失敗します。 |

| オプション | 説明 |
| --- | --- |
| **-ep** | 暗号化キーの入力を求めるプロンプトを表示するよう指定します。このオプションを指定すると、暗号化キーを入力するためのウィンドウが表示されます。クリアテキストでは暗号化キーを見ることができないようにすることで、高いセキュリティが得られます。高度に暗号化されたデータベースの場合、-ek または -ep のどちらかを指定します。両方同時には指定できません。強力に暗号化されたデータベースでは、正しいキーを指定しないとコマンドが失敗します。 |
| **-o** *filename* | 指定したファイルに、出力メッセージを書き込みます。 |
| **-q** | クワイエットモードで実行します (メッセージを表示しません)。このオプションを指定する場合、-y も指定しないと操作は失敗します。 |
| **-y** | 確認メッセージを表示することなく、各ファイルを削除します。-q を指定する場合、-y も指定しないと操作は失敗します。 |

**備考**

消去ユーティリティを使って、データベースファイルと、それに関連するトランザクションログを消去できます。または、トランザクションログファイルやトランザクションログミラーファイルを消去できます。すべてのデータベースファイルとトランザクションログファイルに読み込み専用のマークを付けて、データベースが突然損傷を受けたり、データベースファイルが不用意に削除されないようにします。

*database-file* は、データベースファイルまたはトランザクションログファイルです。ファイル名は、拡張子も含めてすべて指定してください。データベースファイルを指定すると、関連するトランザクションログファイルが (ミラーもある場合はそれも含めて) 消去されます。

**注意**
消去ユーティリティは DB 領域を消去しません。DB 領域を消去したい場合、DROP DATABASE 文を使用するか、または Sybase Central の**データベース消去ウィザード**を使用します。「DROP DBSPACE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

**データベース消去ウィザード**では、DB 領域やトランザクションログファイルを消去することもできます。「データベースの消去」140 ページを参照してください。

他の DB 領域を参照するデータベースファイルを削除しても、DB 領域ファイルが自動的に削除されることはありません。DB 領域ファイルを手動で削除したい場合は、ファイルを読み込み専

用から書き込み可能に変更し、次にファイルを 1 つずつ削除します。別の方法として、DROP DATABASE 文を使用して、データベースと関連する DB 領域ファイルを消去することもできます。

データベースファイルを消去すると、関連するトランザクションログとトランザクションミラーも削除されます。トランザクションログミラーも保有しているデータベースのトランザクションログを消去しても、ミラーは削除されません。

このユーティリティの使用中に、消去するデータベースを起動しないでください。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』を参照してください。

# ファイル非表示ユーティリティ (dbfhide)

単純暗号化を使用して、設定ファイルと初期化ファイルの内容を非表示にします。

### 構文

**dbfhide** [ *options* ] *original-configuration-file encrypted-configuration-file*

| オプション | 説明 |
|---|---|
| **-q** | クワイエットモードで実行します (メッセージを表示しません)。 |
| **-w** | Microsoft から提供されている暗号化 API を使用してファイルを暗号化します。ファイルは、SQL Anywhere データベースサーバー、またはファイルが暗号化されたのと同じコンピューター上のツールによってのみ使用されます。ファイルを暗号化したユーザーのみがファイルを使用できます。このオプションは Windows でのみサポートされています。 |
| **-wm** | Microsoft から提供されている暗号化 API を使用してファイルを暗号化します。ファイルは、SQL Anywhere データベースサーバー、またはファイルが暗号化されたのと同じコンピューター上のツールによってのみ使用されます。ファイルが暗号化されたコンピューター上のどのユーザーでも、ファイルの読み込みが可能です。このオプションは Windows でのみサポートされています。 |
| *original-configuration-file* | 元のファイルの名前を指定します。 |

| オプション | 説明 |
| --- | --- |
| *encrypted-configuration-file* | 難読化された新しいファイルの名前を指定します。 |

### 備考

このユーティリティは、設定ファイルからオプションを読み込む ***@data*** パラメーターを受け入れません。

一部のユーティリティでは、コマンドラインオプションを保存するために設定ファイルが使用されます。これらのオプションにパスワードを含めることができます。デフォルトでは、dbfhide は指定されたファイルで単純な暗号化を使用します。

暗号化されたファイルの内容を変更したり、ファイルから暗号化を削除したりすることはできません。暗号化されたファイルを変更するには、ファイルを置き換える必要があります。したがって、設定ファイルのコピーを作成し、そのコピーを暗号化することをおすすめします。元のファイルは別の場所に保存して、コピーの作成に再使用できるようにします。暗号化された設定ファイルを置き換えるには、元のファイルを使用してコピーを作成し、コピーを変更して暗号化し、古い暗号化ファイルを新しいファイルに置き換えます。

-w または -wm オプションを指定すると、ファイルは Microsoft が提供する API を使用して暗号化されます。API は、ファイルの暗号化に使用された暗号化キーを維持します。暗号化キーは、ユーザーまたはコンピューターのログインにリンクされます。

### 参照


- 「設定ファイルとデータベースサーバーの起動オプション」152 ページ
- 「データのセキュリティ」1379 ページ
- 「設定ファイル」909 ページ
- 「設定ファイルでの条件付き解析」910 ページ
- 「.ini ファイルの内容の非表示」465 ページ
- 「単純暗号化の使用」1397 ページ


### 例

パーソナルデータベースサーバーとサンプルデータベースを開始する設定ファイルを作成します。また、キャッシュを 10 MB に設定し、パーソナルサーバーのこのインスタンスの名前を *Elora* にします。次のように設定ファイルを作成します。

```
# Configuration file for server Elora
-n Elora
-c 10M
"%SQLANY12%¥demo.db"
```

(先頭に # がある行はコメントとして処理されます。)

ファイルに *sample.txt* という名前を付けます。この設定ファイルを使用してデータベースを開始する場合は、コマンドラインで次のように指定します。

```
dbeng12 @sample.txt
```

ここで、単純暗号化を設定に追加します。

```
dbfhide sample.txt encrypted_sample.txt
```

encrypted_sample.txt ファイルを使用してデータベースを開始します。

```
dbsrv12 @encrypted_sample.txt
```

# ヒストグラムユーティリティ (dbhist)

ヒストグラムを Microsoft Excel チャートに変換します。これには、述部の選択性に関する情報が含まれます。

### 構文

**dbhist** [ *options* ] **-t** *table-name* [ *excel-output-filename* ]

| オプション | 説明 |
|---|---|
| *@data* | 指定された環境変数または設定ファイルからオプションを読み込みます。「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル非表示ユーティリティを使用して、設定ファイルの内容を読みにくくすることができます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |
| **-c** *options* | 接続パラメーターを指定します。「接続パラメーター」320 ページを参照してください。 |
| **-n** *colname* | ヒストグラムを関連付けるカラムの名前を指定します。カラムを指定しない場合は、テーブル内のヒストグラムを持っているすべてのカラムが返されます。 |
| **-t** *table-name* | チャートを生成するテーブルまたはマテリアライズドビューの名前を指定します。 |
| **-u** *owner* | テーブルまたはマテリアライズドビューの所有者を指定します。 |
| *excel-output-name* | 生成される Microsoft Excel ファイルの名前を指定します。名前を指定しない場合は、Microsoft Excel から名前の入力を求める **[名前を付けて保存]** ダイアログが表示されます。 |

## 備考

ヒストグラムは ISYSCOLSTAT システムテーブルに格納され、sa_get_histogram ストアドプロシージャーを使用して取り出せます。ヒストグラムユーティリティは、ヒストグラムを Microsoft Excel チャートに変換します。これには、述部の選択性に関する情報が含まれます。ヒストグラムユーティリティ (dbhist) は Windows でのみ使用できます。このユーティリティを使用するためには、コンピューターに Microsoft Excel 97 以降がインストールされている必要があります。

たとえば、統計が最近削除された場合は、テーブルまたはマテリアライズドビューに対する統計 (ヒストグラムを含む) が存在しないことがあります。この場合、ヒストグラムユーティリティによって、「**ヒストグラムにデータが含まれていないため、アボートします。**」というメッセージが表示されます。このような場合は、統計情報を作成してから、再度ヒストグラムユーティリティを実行する必要があります。テーブルまたはマテリアライズドビューの統計を作成するには、CREATE STATISTICS 文を実行します。

文字列カラムに対する述部の選択性を決定するには、ESTIMATE または ESTIMATE_SOURCE 関数を使用してください。文字列カラムからヒストグラムを取り出そうとすると、sa_get_histogram とヒストグラムユーティリティがエラーを生成します。

シート名にはカラム名が使用されます。カラム名は、25 文字目以降がトランケートされ、¥、/、?、*、[、]、: (Microsoft Excel では使用できない文字) はアンダースコア (_) に置き換えられます。チャート名は chart で始まり、上記と同じ命名規則が適用されます。名前が重複した場合は (文字の置換、トランケート、カラム名が chart で始まることなどによって)、重複した名前は使用できないことを示す Microsoft Excel エラーが発生します。ただし、スプレッドシートは作成され、以前のバージョンで作成された名前 (Sheet1、Chart1 など) が付けられます。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

sa_get_histogram ストアドプロシージャーを使用して、ヒストグラムを取得することもできます。

## 参照

- 「CREATE STATISTICS 文」『SQL Anywhere サーバー SQL リファレンス』
- 「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』
- 「sa_get_histogram システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「選択性推定ソース」『SQL Anywhere サーバー SQL の使用法』
- 「オプティマイザーの推定と統計」『SQL Anywhere サーバー SQL の使用法』
- 「ESTIMATE 関数 [その他]」『SQL Anywhere サーバー SQL リファレンス』
- 「ESTIMATE_SOURCE 関数 [その他]」『SQL Anywhere サーバー SQL リファレンス』

## 例

次のコマンド (全体を 1 つの行に入力) は、データベース *demo.db* にあるテーブル SalesOrderItems のカラム ProductID の Microsoft Excel チャートを生成し、それを *histogram.xls* として保存します。

```
dbhist -c "UID=DBA;PWD=sql;DBF=%SQLANYSAMP12%¥demo.db" -n ProductID -t SalesOrderItems
histogram.xls
```

次の文は、テーブル SalesOrders 内にヒストグラムがあるすべてのカラムを対象とするチャートを生成します。サンプルデータベースがすでに起動されていることを前提とします。この文では、UID=DBA と PWD=sql を使用して接続も行います。出力ファイル名を指定していないので、Microsoft Excel から入力するように要求されます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
dbhist -t SalesOrders -c "UID=DBA;PWD=sql"
```

# 情報ユーティリティ (dbinfo)

指定したデータベースに関する情報を表示します。

## 構文

**dbinfo** [ *options* ]

| オプション | 説明 |
|---|---|
| *@data* | 指定された環境変数または設定ファイルからオプションを読み込みます。「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル非表示ユーティリティを使用して、設定ファイルの内容を読みにくくすることができます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |
| **-c** *"keyword=value; ..."* | 接続パラメーターを指定します。「接続パラメーター」320 ページを参照してください。<br><br>有効なユーザー ID は情報ユーティリティを実行できますが、ページの使用状況に関する統計を取得するには DBA 権限が必要です。 |
| **-o** *filename* | 指定したファイルに、出力メッセージを書き込みます。 |
| **-q** | クワイエットモードで実行します (メッセージを表示しません)。 |
| **-u** | システムテーブルやユーザー定義のテーブルを含むすべてのテーブルとマテリアライズドビューの使用状況とサイズに関する情報を表示します。<br><br>他のユーザーがデータベースに接続しておらず、DBA 権限がある場合にのみ、ページ使用状況に関する統計情報を要求できます。ページ使用状況は、sa_table_page_usage システムプロシージャーを使用して取得されます。 |

## 備考

dbinfo ユーティリティを使用すると、データベースに関する情報が表示されます。データベースの名前、トランザクションログファイルまたはログミラーの名前、ページサイズ、照合名とラベ

ル、テーブル暗号化が有効であるかどうかなどの情報がレポートされます。必要に応じて、テーブルの使用状況に関する統計とその詳細を含めることもできます。

dbinfo ユーティリティを使用して、ディスク上のテーブルのサイズを判断できます。そのためには、次のようなコマンドを実行します。

```
dbinfo -u -c "UID=DBA;PWD=sql;DBF=%SQLANYSAMP12%¥demo.db"
```

結果は、データベース内の各テーブルに含まれるデータを保持するために使用されているページ数 (Pages) と、それらのページの使用率 (%used) を示します。すべてのテーブルについて、ページ数にデータベースページのサイズを掛け、その結果に %used を掛けて、そのテーブル用に使用されている領域を判断できます。

また、次の操作を行うことで、データベースに関する情報をさらに取得できます。

- DB_PROPERTY システム関数を使用した個々のデータベースプロパティの問い合わせ。
- sa_db_properties システムプロシージャーを使用したすべてのデータベースプロパティの問い合わせ。
- ping ユーティリティ (dbping) での -pd オプションの使用。-pd を指定すると、指定された各データベースプロパティの値が dbping から返されます。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

**参照**

- 「DB_PROPERTY 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』
- 「sa_db_properties システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「Ping ユーティリティ (dbping)」994 ページ
- 「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』

# 初期化ユーティリティ (dbinit)

新しいデータベースを作成します。

**構文**

**dbinit** [ *options* ] *new-database-file*

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| *@data* | 指定された環境変数または設定ファイルからオプションを読み込みます。 「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル非表示ユーティリティを使用して、設定ファイルの内容を読みにくくすることができます。 「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |
| **-a** | CHAR データ型または NCHAR データ型で Unicode 照合アルゴリズム (UCA ) を使用している場合 (「-z」 と「-zn」 を参照)、文字列比較の際に文字のアクセント記号の違いを考慮します (たとえば、e は é より小さいと見なします)。UCA 照合を使用して作成された日本語のデータベースを除き、デフォルトではアクセント記号は無視されます (e は é と等しいと見なされます)。すべての基本文字 (アクセントと大文字／小文字の区別を取り除いた文字) が等しい場合は、アクセントが左から右へ比較されます。<br><br>日本語のデータベースを作成する場合の UCA 照合では、デフォルトでアクセント記号が区別されます。つまり、アクセント記号が考慮されます。「Unicode 照合アルゴリズム (UCA)」489 ページを参照してください。 |
| **-af** | CHAR データ型または NCHAR データ型で UCA を使用している場合 (「-z」 と「-zn」 を参照)、文字列比較の際に文字のアクセント記号の違いを考慮します (たとえば、e は é より小さいと見なします)。デフォルトでは、アクセント記号は無視されます (e は é と等しいと見なされます)。すべての基本文字 (アクセント記号を取り除いた文字) が等しい場合は、フランス語の規則に従ってアクセント記号が右から左へ比較されます。<br><br>「Unicode 照合アルゴリズム (UCA)」489 ページを参照してください。 |

| オプション | 説明 |
| --- | --- |
| **-b** | データベースにブランクを埋め込みます。<br><br>SQL Anywhere は、文字列について、可変長であり VARCHAR ドメインを使用して格納されている文字列と同じ扱いで、すべての文字列を比較します。これには、固定長の CHAR カラムまたは NCHAR カラムの文字列比較も含まれます。また、値がデータベースに格納されている場合、SQL Anywhere は後続ブランクのトリムや埋め込みは行いません。<br><br>デフォルトでは、SQL Anywhere はブランクを意味のある文字として扱います。したがって、値 'a ' (文字 'a' と、後続の 1 つのブランク) は、単一文字の文字列 'a' と等しくありません。不等号比較の照合でも、ブランクは、ブランク以外の文字として扱われます。<br><br>ブランク埋め込みが有効である場合 (dbinit -b オプション)、文字列比較のセマンティックは ANSI/ISO SQL 標準と一層密接になります。ブランク埋め込みが有効であると、SQL Anywhere はどのような比較であっても後続ブランクを無視します。<br><br>上に挙げた例では、ブランクを埋め込まれたデータベースで 'a ' を 'a' に対して等号比較すると、TRUE が返されます。ブランクを埋め込まれたデータベースでは、固定長文字列の値は、アプリケーションによってフェッチされたときにブランクで埋め込まれます。このような文字の割り当てが行われたときにアプリケーションが文字列のトランケーション警告を受け取るかどうかは、ansi_blanks 接続オプションによって制御されます。「ansi_blanks オプション」592 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| **-c** | 比較や文字列操作をするときに、すべての値で大文字と小文字を区別します。大文字と小文字を区別するデータベースであっても、データベースの識別子については大文字と小文字は区別されません。<br><br>UCA 照合を使用して作成された日本語のデータベースを除き、デフォルトではすべての比較において大文字と小文字が区別されません。日本語のデータベースを作成する場合の UCA 照合では、デフォルトで大文字と小文字が区別されます。<br><br>QAnywhere サーバーストアとして使用されるデータベースでは、大文字と小文字を区別する必要があります。<br><br>このオプションは、ISO/ANSI SQL 標準との互換性を保つために用意されています。 |

| オプション | 説明 |
| --- | --- |
| **-dba** [*DBA-user*][ ,*pwd* ] | DBA ユーザー ID とパスワードを指定します。データベースの DBA ユーザーに新しい名前を指定すると、そのデータベースにユーザー DBA として接続することはできなくなります。DBA データベースユーザーに別のパスワードを指定することもできます。パスワードを指定しない場合は、デフォルトパスワードの sql が使用されます。このオプションを指定しない場合は、パスワードが sql のデフォルトユーザー ID DBA が作成されます。<br><br>次のコマンドは、いずれも DBA ユーザー名を testuser、デフォルトパスワードを sql としてデータベースを作成します。<br><br>dbinit -dba testuser mydb.db<br><br>dbinit -dba testuser, mydb.db<br><br>次のコマンドは、パスワードが mypwd であるデフォルトユーザー ID DBA を使用します。<br><br>dbinit -dba ,mypwd mydb.db<br><br>次のコマンドは、DBA ユーザーをパスワードが mypwd である user1 に変更します。<br><br>dbinit -dba user1,mypwd mydb.db<br><br>パスワードには 7 ビット ASCII 文字を使用することをおすすめします。それ以外の文字を使用すると、サーバーがクライアントの文字セットを UTF-8 に変換できない場合、パスワードが機能しないことがあります。 |
| **-dbs** *size*[ **k** \| **m** \| **g** \| **p** ] | データベースの領域を事前に割り付けます。データベースが使用する領域の事前割り付けを行うと、データベースがあるドライブの空き領域が不足する危険性を小さくすることができます。また、データベースサイズを拡大する操作は時間を要するため、それが必要となる前にデータベースに保存できるデータの量を増やすことがパフォーマンスの向上につながります。<br><br>デフォルトでは、*size* の値はバイト単位となります。単位をキロバイト、メガバイト、ギガバイト、またはページで指定するには、それぞれ **k**、**m**、**g**、**p** を使用します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| **-ea** *algorithm* | データベース暗号化またはテーブル暗号化 (-et) で使用する暗号化アルゴリズムを指定します。単純暗号化の場合は、(-ek や -ep を指定するのではなく) -ea simple を指定します。単純暗号化はデータベースの難読化に相当し、データベースファイルが不用意に直接アクセスされた場合にデータが表示されないようにすることだけを目的としています。これにより第三者は、ディスクユーティリティを使用してファイルを表示し、データベースのデータを解読することが困難になります。<br><br>セキュリティを強化するには、128 ビットの場合は AES、256 ビットの場合は AES256 の強力な暗号化を指定します。FIPS 認定の暗号化を使用するには、128 ビットの場合は AES_FIPS、256 ビットの場合は AES256_FIPS をそれぞれ指定してください。強力な暗号化を使用するためには、-ek または -ep オプションも指定する必要があります。「強力な暗号化」1397 ページを参照してください。<br><br>暗号化されていないデータベースを作成するには、-ea none を指定するか、-ea オプションを指定しません (-et、-ep、-ek オプションも指定しません)。<br><br>-ea オプションを指定しない場合、デフォルトの動作は次のようになります。<br><br>● -ea none (-ek、-ep、-et が指定されない場合)<br>● -ea AES (-ek または -ep が指定される場合) (-et の指定とは無関係)<br>● -ea simple (-ek または -ep を指定せずに -et を指定する場合)<br><br>アルゴリズム名の大文字と小文字は区別されません。<br><br>Windows Mobile では、ARM プロセッサー用に AES_FIPS および AES256_FIPS アルゴリズムのみがサポートされています。<br><br>次のコマンドは、強力に暗号化されたデータベースを作成し、暗号化キーとアルゴリズムを指定します。<br><br>dbinit -ek "0kZ2o56AK#" -ea AES_FIPS "myencrypteddb.db" |

| オプション | 説明 |
| --- | --- |
| | ただし、ファイル圧縮ユーティリティを使用する場合、暗号化を実行したデータベースは、暗号化されていないデータベースほどには圧縮できません。<br><br>**注意**<br>別途ライセンスが必要な必須コンポーネント<br><br>ECC 暗号化と FIPS 認定の暗号化には、別途ライセンスが必要です。強力な暗号化テクノロジはすべて、輸出規制対象品目です。<br><br>「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。 |
| **-ek** *key* | コマンドに暗号化キーを直接指定することで、強力に暗号化されたデータベースを作成するように指定します。-ek オプションは、-ea オプションによって任意で指定された AES アルゴリズムとともに使用します。-ea オプションを指定しないで -ek オプションを指定すると、デフォルトで AES が使用されます。<br><br>このオプションを -et とともに指定すると、データベースは暗号化されません。この場合、テーブル暗号化が有効になります。「テーブル暗号化」1405 ページを参照してください。<br><br>**警告**<br>強力な暗号化が適用されたデータベースの場合、キーのコピーは必ず安全な場所に保管してください。暗号化キーがわからなくなった場合は、Sybase 製品の保守契約を結んでいるテクニカルサポートに依頼してもデータにはアクセスできません。アクセスできなくなったデータベースは、廃棄して、新しくデータベースを作成する必要があります。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| **-ep** | ウィンドウに暗号化キーを入力することで、強力に暗号化されたデータベースを作成するように指定します。クリアテキストでは暗号化キーを見ることができないようにすることで、高いセキュリティが得られます。<br><br>暗号化キーは、正確に入力されたことを確認するために 2 回入力してください。キーが一致しない場合は、初期化は失敗します。<br><br>このオプションを -et とともに指定すると、データベースは暗号化されません。この場合、テーブル暗号化が有効になります。<br><br>詳細については、「強力な暗号化」1397 ページを参照してください。 |
| **-et** | -ea オプションで指定した暗号化アルゴリズム (およびキー) を使用したテーブル暗号化を有効にします。このオプションは、データベース全体を暗号化しないで、暗号化されたテーブルを作成する場合に使用します。-et を -ek または -ep とともに指定し、-ea を指定しないと、デフォルトで AES アルゴリズムが使用されます。-et だけを指定した場合は、単純暗号化が使用されます。<br><br>テーブル暗号化を有効にしただけで、テーブルが暗号化されるわけではありません。データベースを作成した後で、テーブルを個別に暗号化する必要があります。「テーブルの暗号化」1407 ページを参照してください。<br><br>テーブルの暗号化が有効な場合、暗号化されたテーブルのテーブルページ、関連するインデックスページ、テンポラリファイルのページ、暗号化されたテーブルのトランザクションを含むトランザクションログページが暗号化されます。<br><br>次の例は、キー abc と暗号化アルゴリズム AES_FIPS を使用して、強力なテーブル暗号化が有効になっているデータベース *new.db* を作成します。<br><br>dbinit -et -ek abc -ea AES_FIPS new.db |

| オプション | 説明 |
| --- | --- |
| **-i** | jConnect システムオブジェクトをデータベースから除外します。jConnect JDBC ドライバーを使用してシステムカタログ情報にアクセスするには、jConnect カタログサポートをインストールする必要があります (デフォルトでインストールされます)。このオプションを指定した場合でも、システム情報にアクセスしないかぎり、JDBC を使用できます。必要に応じて、Sybase Central または ALTER DATABASE 文を使用して、jConnect サポートを後から追加することもできます。<br><br>詳細については、「jConnect システムオブジェクトのデータベースへのインストール」『SQL Anywhere サーバー プログラミング』を参照してください。<br><br>Windows Mobile で使用するデータベースを作成する場合は、「Windows Mobile の jConnect」417 ページを参照してください。 |
| **-k** | SYSCOLUMNS ビューと SYSINDEXES ビューを作成しません。デフォルトでは、データベース作成機能は、Watcom SQL (このソフトウェアのバージョン 4 以前) で使用可能なシステムテーブルとの互換性を保つために、ビュー SYS.SYSCOLUMNS と SYS.SYSINDEXES を生成します。これらのビューは、Sybase Adaptive Server Enterprise の互換ビュー dbo.syscolumns と dbo.sysindexes と競合します。 |
| **-l** | 推奨する照合順をリストした後、停止します。データベースは作成されません。使用可能な照合順のリストは、Sybase Central の**データベース作成ウィザード**に自動的に表示されます。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| **-le** | 使用可能な文字セットエンコードをリストした後、停止します。データベースは作成されません。文字セットエンコードは、1 つ以上のラベルによって識別されます。エンコードの識別に使用できる文字列があります。表示される各テキスト行には、エンコードラベルと、エンコードの識別に使用できる代替ラベルがリストされます。これらのラベルは、SA (SQL Anywhere ラベル)、IANA (Internet Assigned Numbers Authority)、MIME (Multipurpose Internet Mail Extensions)、ICU (International Components for Unicode)、JAVA、または ASE (Adaptive Server Enterprise) の共通カテゴリに分類されます。<br><br>代替ラベルを含めて、文字セットエンコードのリストを参照するには、-le+ オプションを指定します。<br><br>dbinit ユーティリティが文字セットエンコードをレポートするとき、必ず、ラベルの SQL Anywhere バージョンもレポートします。たとえば、次のコマンドを使用すると、CHAR データ型の文字セットエンコード 1250 のレポートが行われます。<br><br>dbinit -ze cp1250 -z uca test.db |
| **-m** *filename* | トランザクションログミラーを作成します。トランザクションログミラーはトランザクションログと同一のコピーで、通常は別のデバイスで管理され、データを確実に保護しています。デフォルトでは、SQL Anywhere はトランザクションログミラーを使用しません。 |
| **-n** | トランザクションログのないデータベースを作成します。トランザクションログを使用しないデータベースを作成すると、ディスク領域を節約できますが、コミットごとにチェックポイントが発生するので、パフォーマンスが低下する可能性があります。また、トランザクションログを使用しないで実行していた場合、データベースが壊れたときにデータを回復できません。ただし、トランザクションログはデータレプリケーションに必要です。メディア障害またはシステム障害の発生時に、データベース情報のセキュリティを強化します。 |

| オプション | 説明 |
| --- | --- |
| **-o** *filename* | 指定したファイルに、出力メッセージを書き込みます。 |
| **-p** *page-size* [ **k** ] | データベースのページサイズを指定します。データベースのページサイズには、2048、4096、8192、16384、32768 バイトのいずれかを指定できます。デフォルトは 4096 バイトです。**k** を使用して、キロバイトの単位を指定します (-p 4k など)。<br><br>大規模なデータベースでは、より大きなページサイズの方が有利です。たとえば、テーブルのスキャンでは一度にページ全体が読み込まれるため、通常、必要な I/O 操作の回数は少なくなります。ただし、大きなページサイズには、より多くのメモリが必要です。ページサイズを選択するときは、パフォーマンステスト (およびテスト全般) を実行することを強くおすすめします。そして、満足できる結果を得られた最小のページサイズを選択します。ほとんどのアプリケーションでは、16 KB または 32 KB のページサイズはおすすめしません。常に十分なデータベースサーバーキャッシュの確保が可能であり、メモリとディスク領域のパフォーマンス特性に対するトレードオフが調査済みである場合以外は、運用システムで 16 KB または 32 KB のページサイズは使用しないでください。多数のデータベースを同じサーバーで起動する場合は、適切なページサイズを選択してください。<br><br>次の項を参照してください。<br><br>● 「適切なページサイズの使用」『SQL Anywhere サーバー SQL の使用法』 |
| **-q** | クワイエットモードで実行します (メッセージを表示しません)。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| **-s**[ - ] | グローバルチェックサムを追加します (チェックサムはデータベースページごとに追加されます)。デフォルトでは、このオプションはオンになります。チェックサムは、データベースページがディスク上で変更されたかどうかを判断するために使用します。グローバルチェックサムを有効にしてデータベースを作成した場合、チェックサムはページがディスクに書き込まれる直前に計算されます。そのページが次にディスクから読み出されるときに、ページのチェックサムが再計算されて、ページに保存されているチェックサムと比較されます。チェックサムが異なる場合は、ディスク上のページが変更されているか、破損しており、エラーが発生します。-s オプションの値を指定したかどうかにかかわらず、重要なデータベースページにはデータベースサーバーによって必ずチェックサムが追加されます。<br><br>Windows Mobile またはストレージデバイス (リムーバブルドライブなど) 上で実行されているデータベースでは、データベースの破損を速やかに検出できるように、自動的にチェックサムが有効になります。<br><br>グローバルチェックサムを無効にしてデータベースが作成された場合でも、START DATABASE 文の -wc オプションを指定すると、ページの書き込み時にチェックサムをページに追加できます。<br>「チェックサムを使用した破損の検出」1120 ページを参照してください。 |

| オプション | 説明 |
| --- | --- |
| **-t** *transaction-log-name* | トランザクションログファイルの名前を指定します。トランザクションログは、使用しているアプリケーションに関わらず、すべてのユーザーが行った変更をデータベースサーバーがその中に記録するファイルです。トランザクションログはバックアップとリカバリ、データレプリケーションで重要な役割を果たします。トランザクションログのファイル名にパスがない場合、トランザクションログはデータベースファイルと同じディレクトリに保存されます。-t または -n を指定しないで dbinit を実行すると、データベースファイルと同じファイル名のトランザクションログが作成されますが、拡張子は *.log* になります。<br><br>詳細については、「トランザクションログ」127 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| **-z** *coll* [ *collation-tailoring-string* ] | データベースの照合順を指定します。照合順は、文字データ型 (CHAR、VARCHAR、LONG VARCHAR) のソートと比較に使用されます。照合は、使用されるエンコード (文字セット) に文字の比較と順序付けに関する情報をもたらすものです。照合は慎重に選択してください。データベースの作成が完了した後で照合を変更するためには、データベースをアンロードし、再ロードする必要があります。照合が指定されていない場合、SQL Anywhere はオペレーティングシステムの言語と文字セットに基づいて照合を選択します。次の項を参照してください。<br><br>● 「照合の考慮事項」496 ページ<br>● 「推奨文字セットと照合」506 ページ<br>● 「サポートされている照合と代替照合」492 ページ<br><br>オプションで、文字列のソートや比較を詳細に制御することを目的に、照合の適合化オプション (*collation-tailoring-string*) を指定できます。これらのオプションは、キーワード=値 の形式で、カッコで囲んで指定して、その後ろに照合名を記述します。次に例を示します。<br><br>dbinit -c -z uca(locale=es;case=LowerFirst) spanish2.db<br><br>「照合の適合化オプション」496 ページを参照してください。<br><br>*collation-tailoring-string* に指定する大文字小文字とアクセント記号の設定は、dbinit (-c、-a、-af) の大文字小文字とアクセント記号の設定よりも優先されます (両方を指定した場合)。<br><br>**注意**<br>照合の適合化オプションを使用して初期化したデータベースは、10.0.1 より前のデータベースサーバーでは起動できません。 |

| オプション | 説明 |
| --- | --- |
| **-ze** *encoding* | 照合のエンコードを指定します。-z で指定するほとんどの照合には、エンコード (文字セット) と順序付けの両方が定義されています。そのような照合については、-ze を指定する必要はありません。<br><br>-z で指定した照合が Unicode 照合アルゴリズム (UCA ) である場合、-ze では CHAR データ型のエンコードとして UTF-8 またはシングルバイトエンコードを指定できます。デフォルトのエンコードは UTF-8 です。-ze を使用すると、ロケール固有のエンコードを指定し、比較と順序付けに UCA を利用することができます。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| **-zn** *coll* [ *collation-tailoring-string* ] | 各国の文字データ型 (NCHAR、NVARCHAR、LONG NVARCHAR) のソートと比較に使用する照合順を指定します。照合は、使用される UTF-8 エンコード (文字セット) に文字の順序付けに関する情報をもたらすものです。このオプションの値は、UCA (デフォルト) と UTF8BIN です。UTF8BIN は、エンコードが 0x7E を超えるすべての文字のバイナリ順を規定します。dbicu12 と dbicudt12 の DLL がインストールされていない場合、デフォルトの NCHAR 照合は UTF8BIN になります。「照合の考慮事項」496 ページを参照してください。<br><br>オプションで、文字列のソートや比較を詳細に制御することを目的に、照合の適合化オプション (*collation-tailoring-string*) を指定できます。これらのオプションは、キーワード=値 の形式で、カッコで囲んで指定して、その後ろに照合名を記述します。次に例を示します。<br><br>dbinit -c -zn UCA(case=LowerFirst) sens.db<br><br>「照合の適合化オプション」496 ページを参照してください。<br><br>*collation-tailoring-string* に指定する大文字小文字とアクセント記号の設定は、dbinit (-c、-a、-af) の大文字小文字とアクセント記号の設定よりも優先されます (両方を指定した場合)。<br><br>**注意**<br>照合の適合化オプションを使用して初期化したデータベースは、10.0.1 より前のデータベースサーバーでは起動できません。 |

**備考**

初期化するときにいくつかのデータベース属性が指定されます。これらの属性は、データベース全体のアンロード、再初期化、再構築以外の方法では、後から変更することはできません。データベース属性には次のものがあります。

- 大文字と小文字の区別の有無
- アクセント記号の区別
- 句読表記の区別
- 比較における後続ブランクの処理
- ページサイズ
- 文字セットエンコードと照合順
- データベースの暗号化
- テーブル暗号化

たとえば、次のようにして 8192 バイトのページを持つデータベース *test.db* を作成できます。

```
dbinit -p 8192 test.db
```

データベースに **utility_db** という名前を付けることはできません。この名前は、ユーティリティデータベースのために予約されています。「ユーティリティデータベース」135 ページを参照してください。

- ユーザー ID には、次に該当する値を指定できません。
  - 空白スペース、一重引用符、または二重引用符で始まる値
  - 空白スペースで終わる値
  - セミコロンを含む値
- パスワードは大文字と小文字が区別され、次に該当する値は指定できません。
  - 空白スペース、一重引用符、または二重引用符で始まる値
  - 空白スペースで終わる値
  - セミコロンを含む値
  - 256 バイト長を超える値

データベースに大文字小文字またはアクセント記号の区別がある場合は、初期化コマンドに照合の適合化オプションを指定するときに、句読表記の区別でレベル 4 は指定できません。

さらに、初期化するときに、トランザクションログとトランザクションログミラーを使うかどうかを選択することができます。この選択は、トランザクションログユーティリティまたは ALTER DATABASE 文を使用して後で変更できます。

**注意**
別途ライセンスが必要な必須コンポーネント

ECC 暗号化と FIPS 認定の暗号化には、別途ライセンスが必要です。強力な暗号化テクノロジはすべて、輸出規制対象品目です。

「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

データベースは、次の方法を使用して作成することもできます。

●Sybase Central の**データベース作成ウィザード**を使用する。

●Interactive SQL から CREATE DATABASE 文を使用する。

> **注意**
> アプリケーションを配備するときは、dbinit ユーティリティを使用してデータベースを作成するために、パーソナルデータベースサーバー (dbeng12) が必要です。パーソナルデータベースサーバーは、その他のデータベースサーバーが実行されていない場合にローカルコンピューターで Sybase Central からデータベースを作成する場合にも必要です。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

### 参照


- 「データベースの作成 (Sybase Central)」 118 ページ
- 「CREATE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』


### 例

次のコマンドを実行すると、大文字と小文字を区別するデータベースである *spanish.db* が作成されます。このデータベースでは、非 NCHAR データに対して 1252spa 照合が使用されます。NCHAR データには、UCA 照合が指定されます。この場合、ロケールは es であり、最初に小文字がソートされます。

```
dbinit -c -z 1252spa -zn uca(locale=es;case=LowerFirst) spanish.db
```

## Interactive SQL ユーティリティ (dbisql)

SQL 文を実行し、データベースに対してスクリプトファイルを実行します。

### 構文

**dbisql -c** "*connection-string*" [ *options* ] [ *dbisql-statement* | *dbisql-script-file* ]

*dbisql-statement* : 1 つの SQL 文またはコマンドデリミターによって区切られる一連の SQL 文。

| オプション | 説明 |
| --- | --- |
| **@***data* | 指定された環境変数または設定ファイルからオプションを読み込みます。<br><br>同じ名前の環境変数と設定ファイルが存在する場合は、環境変数が使用されます。「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル難読化ユーティリティを使用して、設定ファイルの内容を難読化できます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |
| **-c** *"keyword=value; ..."* | 接続パラメーターを指定します。Interactive SQL が接続できない場合は、接続パラメーターを入力するウィンドウが表示されます。ユーザー ID とパスワードの両方を指定しなかった場合は、デフォルトの UID **DBA** と PWD **sql** が使用されます。「接続パラメーター」320 ページを参照してください。 |
| **-d** *delimiter* | コマンドデリミターを指定します。デリミターを囲む引用符は省略可能ですが、コマンドシェル自体がデリミターを特別な方法で解釈するときは必ず指定します。<br><br>このオプションは、command_delimiter オプションの設定を上書きします。「command_delimiter オプション [Interactive SQL]」887 ページを参照してください。 |
| **-d1** | ユーザーが明示的に実行するすべての文をコマンドウィンドウ (STDOUT) にエコーします。これによって、SQL スクリプトのデバッグ、または Interactive SQL が長文の SQL スクリプトを処理しているときに有用なフィードバックが提供されます。(最後の文字は数値の 1 であり、L の小文字ではありません)。このオプションは、Interactive SQL をコマンドラインプログラムとして実行している場合のみ使用できます。 |
| **-datasource** *DSN-name* | 接続先の ODBC データソースを指定します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| **-f** *filename* | [SQL 文] ウィンドウ枠で *filename* というファイルを (実行しないで) 開きます。<br><br>-f オプションが指定されている場合、-c オプションは無視されます。つまり、データベースへの接続は確立されません。<br><br>ファイル名は引用符で囲んでも囲まなくても構いませんが、ファイル名にスペースが含まれている場合は必ず引用符で囲む必要があります。<br><br>そのファイルが存在しない場合、またはファイルではなく実際にはディレクトリである場合は、Interactive SQL がエラーメッセージを出力して終了します。<br><br>ファイル名に完全なドライブとパスの仕様が含まれていないときは、現在のディレクトリが基準として想定されます。<br><br>このオプションは、Interactive SQL をウィンドウベースのアプリケーションとして実行している場合のみサポートされます。 |
| **-host** *hostname* | データベースサーバーを実行するコンピューターの*ホスト名*または IP アドレスを指定します。現在のコンピューターを表す localhost という名前を使用できます。 |

| オプション | 説明 |
| --- | --- |
| **-nogui** | Interactive SQL をコンソールアプリケーションとして実行します。このとき、ウィンドウベースのユーザーインターフェイスは使用しません。これは、バッチ処理に便利です。<br><br>*dbisql-statement* または *dbisql-script-file* のいずれかを指定する場合、-nogui が使用されます。<br><br>このモードのとき、Interactive SQL は、処理の成功または失敗をプログラム終了コードを設定することで示します。Windows オペレーティングシステムでは、プログラム終了コードに対して環境変数 ERRORLEVEL が設定されます。「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』を参照してください。 |
| **-onerror** { **continue** \| **exit** } | スクリプトファイルから文を読み出し中にエラーが起こった場合の事象を制御します。これは、Interactive SQL をバッチ処理で使用するときに便利です。このオプションは、on_error 設定を上書きします。「on_error オプション [Interactive SQL]」898 ページを参照してください。<br><br>*behavior* には次のいずれかの値を定義できます。<br><br>● **Continue**　エラーは無視され、Interactive SQL は文の実行を継続します。<br><br>● **Exit**　Interactive SQL は終了します。 |
| **-port** *port-number* | SQL Anywhere データベースサーバーが実行されているポート番号を指定します。SQL Anywhere のデフォルトポート番号は 2638 です。 |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| **-q** | 出力メッセージを表示しません。ユーティリティをクワイエットモードで実行するように設定します。これは、文またはスクリプトファイルを使って Interactive SQL を起動したときにのみ便利です。このオプションを指定した場合、エラーメッセージは表示されますが、次の情報は表示されません。<br><br>●警告およびその他の致命的でないメッセージ<br><br>●結果セットの出力 |
| **-ul** | デフォルトとして Ultra Light データベースを指定します。Interactive SQL では、接続するデータベースのタイプに応じて使用可能なオプションがカスタマイズされます。<br><br>Interactive SQL では、デフォルトで SQL Anywhere データベースに接続すると見なされます。-ul オプションを指定すると、デフォルトが Ultra Light データベースに変更されます。デフォルトに設定されているデータベースのタイプに関係なく、**[接続]** ウィンドウの **[データベースタイプを変更]** ドロップダウンリストからデータベースのタイプを選択することで、SQL Anywhere または Ultra Light のデータベースに接続できます。<br><br>Interactive SQL から Ultra Light データベースへの接続については、「Ultra Light 用 Interactive SQL ユーティリティ (dbisql)」『Ultra Light データベース管理とリファレンス』を参照してください。 |
| **-version** | Interactive SQL のバージョン番号を表示します。Interactive SQL 内から **[ヘルプ]** → **[Interactive SQL について]** をクリックして、バージョン番号を参照することもできます。 |

| オプション | 説明 |
| --- | --- |
| **-x** | 文をスキャンしますが、実行しません。長いスクリプトファイルの構文エラーをチェックする場合に有用です。<br><br>SQL 文と Interactive SQL 文の詳細については、「SQL 言語の要素」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| *dbisql-statement* \| *dbisql-script-file* | SQL 文または指定された *dbisql-script-file* を実行します。<br><br>*dbisql-statement* または *dbisql-script-file* を指定しなかった場合、Interactive SQL は対話型モードになり、コマンドをコマンドウィンドウに入力できます。 |

**備考**

Interactive SQL を使用して、データベースのブラウズ、SQL 文の実行、およびスクリプトファイルの実行を行うことができます。また、次の項目に関するフィードバックも提供します。

- 影響を受けたローの数
- 各文に必要な時間
- クエリの実行プラン
- エラーメッセージ

SQL Anywhere データベースと Ultra Light データベースの両方に接続できます。Ultra Light データベースへの接続方法については、「Ultra Light 用 Interactive SQL ユーティリティ (dbisql)」『Ultra Light データベース管理とリファレンス』を参照してください。

Interactive SQL は、Windows、Solaris、Linux、Mac OS X でサポートされています。http://www.ianywhere.jp/tech/1061806-os_components.html を参照してください。

Windows の場合、実行プログラムが 2 つあります。

1. バッチスクリプトでは、*dbisql.exe* ではなく、*dbisql* または *dbisql.com* を呼び出す必要があります。実行プログラム *dbisql.com* は、コンソールアプリケーションとしてリンクされています。

2. 実行プログラム *dbisql.exe* はウィンドウアプリケーションとしてリンクされており、起動元のコマンドシェルがブロックされません。*dbisql.exe* をバッチファイルから実行した場合、標準出力または標準のエラーファイルに出力が送信されません。

Interactive SQL のデフォルトのエンコードは、default_isql_encoding オプションでも一時的に設定できます。「default_isql_encoding オプション [Interactive SQL]」888 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

INPUT、OUTPUT、または READ 文の ENCODING 句を使用して、ファイルの読み込みまたは書き込みをするときに使用するエンコードを指定できます。次の項を参照してください。

- 「INPUT 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「OUTPUT 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「READ 文 [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』

終了コードは、0 (成功) または 0 以外の値 (失敗) です。0 以外の終了コードが設定されるのは、(SQL 文またはスクリプトファイル名を指定したコマンドラインによって) Interactive SQL をバッチモードで実行した場合だけです。「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』を参照してください。

Interactive SQL で *reload.sql* ファイルを実行する場合は、パラメーターとして暗号化キーを指定する必要があります。READ 文にキーを指定していない場合、Interactive SQL でキーの入力を要求するプロンプトが表示されます。

次の方法で Interactive SQL を開始できます。

- Sybase Central で **[ファイル]** » **[Interactive SQL を開く]** をクリックする。
- **[スタート]** メニューから、**[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[Interactive SQL]** をクリックする。
- コマンドプロンプトで dbisql コマンドを使用する。

### 参照

- 「Interactive SQL の SQL 文」878 ページ
- 「Interactive SQL」840 ページ

### 例

次のコマンドを入力すると、ユーザー ID DBA とパスワード sql で、現在のデフォルトサーバーに対してスクリプトファイル *mycom.sql* が実行されます。スクリプトファイルにエラーがあった場合は、処理は停止します。

```
dbisql -c "UID=DBA;PWD=sql" -onerror exit mycom.sql
```

次のコマンドを入力すると、現在のデフォルトデータベースにユーザーが追加されます。

```
dbisql -c "UID=DBA;PWD=sql" CREATE USER joe IDENTIFIED BY passwd
```

## キーペアジェネレーターユーティリティ (createkey)

Mobile Link エンドツーエンド暗号化で使用する RSA と ECC のキーペアを作成します。

### 構文

**createkey**

### 備考

ECC オブジェクトを作成するには、別途ライセンスを注文する必要があります。「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

createkey を実行する場合は、次の情報を求めるプロンプトが表示されます。

- **[暗号化タイプを選択してください。]** このプロンプトは、ECC 暗号化のライセンスを購入した場合にかぎり表示されます。[RSA] または [ECC] を選択します。

- **[RSA のキー長を入力してください (512 ～ 16384)。]** このプロンプトは、RSA 暗号化を選択した場合にかぎり表示されます。512 ～ 16384 ビットの間で長さを選択できます。

- **[ECC 曲線]** このプロンプトは、ECC 暗号化のライセンスを購入し、かつ ECC 暗号化タイプを選択した場合にかぎり表示されます。ECC 曲線のリストから選択することを求めるプロンプトが表示されます。デフォルトは secp521r1 です。

  曲線名の 3 桁の数値は、曲線のビットサイズを表します。曲線の暗号化強度は、半分のサイズの AES キーとほぼ同等です。たとえば、512 ビットの ECC 曲線は、256 ビットの AES キーと同じ強度になります。

  次の ECC 曲線がサポートされています。

  - sect163k1
  - sect163r2
  - sect233k1
  - sect233r1
  - sect283k1
  - sect283r1
  - sect409k1
  - sect409r1
  - sect571k1
  - sect571r1
  - secp192r1
  - secp224r1
  - secp256r1
  - secp384r1
  - secp521r1

- **[パブリックキーを格納するファイルパスを入力してください。]** 生成された、PEM または DER でコード化されたパブリックキーのファイル名とロケーションを指定します。このファイルは、Mobile Link クライアントで e2ee_public_key プロトコルオプションによって指定されます。「e2ee_public_key」『Mobile Link クライアント管理』を参照してください。

- **[プライベートキーを保存するファイル名を入力してください。]** 生成された、PEM または DER でコード化されたプライベートキーのファイル名とロケーションを指定します。このファイルは、Mobile Link サーバーで e2ee_private_key プロトコルオプションによって指定されます。「-x mlsrv12 オプション」『Mobile Link サーバー管理』を参照してください。

- **[プライベートキーを保護するためのパスワードを入力してください。]** 任意で、プライベートキーを暗号化するために使用するパスワードを指定します。パスワードを指定しない場合、

プライベートキーは暗号化されません。このパスワードは、Mobile Link サーバーで e2ee_private_key_password プロトコルオプションによって指定されます。「-x mlsrv12 オプション」『Mobile Link サーバー管理』を参照してください。

### 参照

- 「エンドツーエンド暗号化」 1426 ページ
- 「e2ee_type」『Mobile Link クライアント管理』 (Mobile Link クライアントネットワークプロトコルオプション)

### 例

次の例では、RSA キーペアを作成します。

```
createkey
SQL Anywhere キーペアジェネレーターバージョン 12.0.1.1304
暗号化タイプを選択してください ((R)SA または (E)CC) : r
RSA のキー長を入力してください (512 ～ 16384) : 2048
キーペア作成中...
パブリックキーを格納するファイルパスを入力してください : rsapublic.pem
プライベートキーを保存するファイル名を入力してください : rsaprivate.pem
プライベートキーを保護するためのパスワードを入力してください : pwd
```

## 言語選択ユーティリティ (dblang)

SQL Anywhere、 Sybase Central、Interactive SQL によって使用される言語を制御するレジストリ設定のレポートと変更を行います。

### 構文

**dblang** [ *options* ] *language-code*

| オプション | 説明 |
|---|---|
| **-m** | HKEY_LOCAL_MACHINE の下のレジストリに、言語コードを書き込みます。 |
| **-q** | クワイエットモードで実行します (メッセージを表示しません)。 |
| **-u** | HKEY_CURRENT_USER の下のレジストリに、言語コードを書き込みます。これがデフォルトのロケーションです。 |

| 言語コード | 言語 |
|---|---|
| EN | 英語 |
| DE | ドイツ語 |

| 言語コード | 言語 |
|---|---|
| ES | スペイン語 |
| FR | フランス語 |
| IT | イタリア語 |
| JA | 日本語 |
| KO | 韓国語 |
| LT | リトアニア語 |
| PL | ポーランド語 |
| PT | ポルトガル語 |
| RU | ロシア語 |
| TW | 中国語 (繁体文字) |
| UK | ウクライナ語 |
| ZH | 中国語 (簡体文字) |

**備考**

-m または -u を指定しないと、言語コードは HKEY_CURRENT_USER の下のレジストリに書き込まれます。-m と -u の両方を指定すると、両方のロケーションに言語コードを書き込むことができます。

言語コードを指定しないで dblang ユーティリティを実行すると、現在の設定がレポートされます。次の設定があります。

- **SQL Anywhere**　この設定は、SQL Anywhere データベースサーバーから情報メッセージとエラーメッセージを配信するときに使用される言語リソースライブラリを制御します。言語リソースライブラリは、*dblgXX12.dll* という形式の名前を持つ DLL です (*XX* は 2 文字の言語コードです)。

  この設定を変更するときは、自分のコンピューター上に適切な言語リソースライブラリがあることを確認してください。

- **Sybase Central と Interactive SQL**　この設定は、Sybase Central と Interactive SQL のユーザーインターフェイス要素を表示するために使用されるリソースを制御します。

言語の変更による高速ランチャーへの影響の詳細については、「高速ランチャーオプション」903 ページを参照してください。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

このユーティリティは、設定ファイルからオプションを読み込む ***@data*** パラメーターを受け入れません。

### 参照

- 「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』
- 「SALANG 環境変数」450 ページ
- 「Language (LANG) 接続パラメーター」351 ページ
- 「管理ツールの言語設定の変更」904 ページ

### 例

次のコマンドは、現在の設定を示すウィンドウを表示します。

```
dblang
```

次のコマンドは、設定をドイツ語に変更し、前の設定と新しい設定を示すウィンドウを表示します。

```
dblang de
```

## ログ変換ユーティリティ (dbtran)

トランザクションログを SQL スクリプトファイルに変換します。

### 構文

データベースサーバーに対して実行する場合。

**dbtran** [ *options* ] **-c** { *connection-string* } **-n** *SQL-script-file*

トランザクションログに対して実行する場合。

**dbtran** [ *options* ] [ *transaction-log* ] [ *SQL-script-file* ]

| オプション | 説明 |
|---|---|
| *@data* | 指定された環境変数または設定ファイルからオプションを読み込みます。「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル非表示ユーティリティを使用して、設定ファイルの内容を読みにくくすることができます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |

| オプション | 説明 |
| --- | --- |
| **-a** | コミットされていないトランザクションをトランザクションログに含めるかどうかを制御します。<br><br>トランザクションログには、最新の COMMIT 以前にすべてのトランザクションによって加えられた変更が含まれます。最新のコミット以後の変更はトランザクションログの中にはありません。<br><br>-a が指定されていない場合、出力ファイルにはコミットされたトランザクションのみが含まれます。-a を指定した場合、トランザクションログにあるコミットされていないトランザクションに続いて、ROLLBACK 文が出力されます。 |
| **-c** *"keyword=value; ..."* | ユーティリティをデータベースサーバーに対して実行する場合に、接続文字列を指定します。「接続パラメーター」320 ページを参照してください。 |
| **-d** | トランザクションを古いものから新しいものへと順に書き出すことを指定します。この機能は、主に、データベースのアクティビティを監査するときに使用されます。dbtran の出力を、データベースに対して適用しないようにしてください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| **-ek** *key* | 強力に暗号化されているデータベースの暗号化キーを指定します。データベースが強力に暗号化されている場合、データベースまたはトランザクションログを使用するには暗号化キーを指定する必要があります。<br><br>強力な暗号化が適用されたデータベースの場合、-ek または -ep のどちらかを指定します。両方同時には指定できません。正しい暗号化キーを指定しないとコマンドが失敗します。<br><br>データベースサーバーに対して -c オプションを使用して dbtran を実行している場合は、-ek オプションではなく、接続パラメーターを使用してキーを指定してください。たとえば、次のコマンドは、データベースサーバー **sample** からデータベース *myenc.db* についてのトランザクションログ情報を取得し、その出力を *log.sql* に保存します。<br><br>dbtran -n log.sql -c "Server=sample;DBF=myenc.db; UID=DBA;PWD=sql;DBKEY=mykey" |
| **-ep** | 暗号化キーを入力するよう要求します。このオプションを指定すると、暗号化キーを入力するためのウィンドウが表示されます。クリアテキストでは暗号化キーを見ることができないようにすることで、高いセキュリティが得られます。<br><br>強力な暗号化が適用されたデータベースの場合、-ek または -ep のどちらかを指定します。両方同時には指定できません。正しい暗号化キーを指定しないとコマンドが失敗します。<br><br>データベースサーバーに対して -c オプションを使用して dbtran を実行している場合は、-ep オプションではなく、接続パラメーターを使用してキーを指定してください。たとえば、次のコマンドは、データベースサーバー **sample** からデータベース *myenc.db* についてのトランザクションログ情報を取得し、その出力を *log.sql* に保存します。<br><br>dbtran -n log.sql -c "Server=sample;DBF=myenc.db; UID=DBA;PWD=sql;DBKEY=mykey" |

| オプション | 説明 |
| --- | --- |
| **-f** | 最終チェックポイント以降に完了したトランザクションだけを出力します。 |
| **-g** | auditing データベースオプションがオンの場合は、監査情報をトランザクションログに追加します。この情報は、このオプションを使用して、出力ファイルにコメントとして含めることができます。「auditing オプション」598 ページを参照してください。<br><br>-g オプションは、-a、-d、-t オプションも暗黙で指定したことになります。 |
| **-ir** *offset1,offset2* | 2 つの指定オフセット間の部分的トランザクションログを出力します。 |
| **-is** *source,...* | 次の 1 つ以上のソース (カンマ区切りのリストで指定) の操作によって変更された、ローに対する操作を出力します。<br><br>● **All**　すべてのロー。これがデフォルト設定です。<br><br>● **SQLRemote**　SQL Remote を使用して変更されたローだけを含みます。SR という短い形式を使用することもできます。<br><br>● **Local**　レプリケートされないローだけを含みます。 |
| **-it** *owner.table,...* | カンマ区切りのリストで指定されたテーブルに対する操作を出力します。各テーブルは、*owner.table* として指定します。 |
| **-j** *date/time* | 指定した日付または時刻より前の、最新のチェックポイント以降のトランザクションだけを変換します。ユーザーは、日付、時刻、日付と時刻を引用符で囲んで引数を指定できます。時刻を省略すると、その日付の開始時刻が使用されます。日付を省略すると、今日の日付が使用されます。日付と時刻には、フォーマット *"YYYY/MMM/DD HH:NN"* を使用できます。 |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| **-k** | エラー検出時に部分的な *.sql* ファイルが消去されるのを防止します。dbtran の実行中にエラーが検出された場合、通常は、部分的なファイルが誤って使用されないように、その時点までに生成された *.sql* ファイルが消去されます。このオプションを指定すると、破損したトランザクションログからトランザクションをサルベージしようとするときに役立ちます。 |
| **-m** | トランザクションログを格納するディレクトリを指定します。このオプションは、-n オプションと一緒に使用してください。 |
| **-n** *filename* | データベースサーバーに対して dbtran ユーティリティを実行するときに、SQL 文を保持する出力ファイルを指定します。 |
| **-o** *filename* | 指定したファイルに、出力メッセージを書き込みます。 |
| **-q** | クワイエットモードで実行します (メッセージを表示しません)。 |
| **-r** | コミットされなかったトランザクションを削除します。これはデフォルトの動作です。 |
| **-s** | UPDATE 文の生成方法を制御します。テーブルにプライマリキーまたはユニークなインデックスがない場合、重複ローが存在する可能性があるため、このオプションを指定しないと、ログ変換ユーティリティは標準ではない FIRST キーワードを使って UPDATE 文を作成します。このオプションを使用すると、FIRST キーワードが省略され、SQL 標準との互換性が保持されます。 |
| **-sr** | SQL Remote がリモートサイトに操作を分配する方法を記述するコメントを生成し、出力ファイルに挿入します。 |

| オプション | 説明 |
| --- | --- |
| **-t** | トリガーをスクリプトファイルに含めるかどうかを制御します。デフォルトでは、トリガーが実行する動作はスクリプトファイルには含まれません。一致するトリガーがデータベースにある場合は、データベースに対してスクリプトファイルを実行すると、トリガーの動作が自動的に実行されます。すファイルが実行されるデータベース内に一致するトリガーがない場合は、トリガーの動作を出力に入れてください。 |
| **-u** *userid,...* | 指定したユーザーのトランザクションログだけが出力されるように制限します。 |
| **-x** *userid,...* | 指定したユーザー以外のトランザクションログが出力されるように制限します。 |
| **-y** | 確認メッセージを表示することなく、既存のスクリプトファイルを置き換えます。-q を指定する場合、-y も指定しないと操作は失敗します。 |
| **-z** | トリガーによって生成されたトランザクションを、コメントとして出力ファイルの中に含めます。 |
| *transaction-log* | 変換するログファイルを指定します。-c または -m オプションとは一緒に使用できません。 |
| *SQL-file* | 変換した情報を含む出力ファイルを指定します。*transaction-log* 専用です。 |

**備考**

dbtran ユーティリティは、トランザクションログ内の情報を取り出して、それらを一連の SQL 文とコメントとして出力ファイルに入れます。このユーティリティは、次の方法で実行できます。

- **データベースサーバーに対して実行**　データベースサーバーに対して実行した場合、dbtran は標準的なクライアントアプリケーションとして動作します。これは、-c オプションの後に指定された接続文字列を使用してデータベースサーバーに接続し、-n オプションによって指定されたファイルに出力を送ります。この方法での実行には、DBA 権限が必要です。

  次のコマンドは、サーバー **demo12** からのログ情報を変換して、出力をファイル *demo.sql* に入れます。

  ```
  dbtran -c "Server=demo12;DBN=demo;UID=DBA;PWD=sql" -n demo.sql
  ```


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

- **トランザクションログファイルに対して実行**　トランザクショログに対して実行した場合、dbtran はトランザクションログファイルに対して直接作用します。ユーザーにこのコマンドを実行する機能を持たせないようにする場合は、トランザクションログファイルを一般的なアクセスから保護してください。

```
dbtran demo.log demo.sql
```

dbtran ユーティリティが実行されると、トランザクションログの初期のログオフセットが表示されます。複数のログファイルが作成される場合、順序を決定するにはこの方法が効果的です。

-c を使用する場合、dbtran はオンライントランザクションログファイルと、オンライントランザクションログファイルと同じディレクトリにあるすべてのオフライントランザクションログファイルを変換しようとします。ディレクトリに複数のデータベース用のトランザクションログファイルが含まれている場合、dbtran がエラーを示す場合があります。この問題を回避するには、各ディレクトリに 1 つのデータベースのみのトランザクションログファイルが含まれていることを確認します。

トランザクションは複数のトランザクションログにまたがる場合があります。トランザクションログファイルに複数のログにまたがるトランザクションが含まれている場合、単一のトランザクションログファイル (たとえば dbtran demo.log) を変換すると、関連しているトランザクションが失われる場合があります。dbtran によって完全なトランザクションを生成するには、ディレクトリ内のトランザクションログファイルで -c または -m オプションを使用します。「トランザクションログが複数の場合のデータベースのリカバリ」1094 ページを参照してください。

次の方法で、ログ変換ユーティリティにアクセスできます。

- Sybase Central の**ログファイル変換ウィザード**を使用する。
- コマンドプロンプトで、dbtran コマンドを入力する。バッチまたはコマンドファイルへの組み込みには、このユーティリティが便利です。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』を参照してください。

# パフォーマンス統計値ユーティリティ (dbstats) (UNIX)

UNIX で SQL Anywhere データベースサーバーのパフォーマンス統計値を返します。

### 構文

**dbstats** [ *options* ] [ *interval* ] [ *iterations* ]

| オプション | 説明 |
| --- | --- |
| *@data* | 指定された環境変数または設定ファイルからオプションを読み込みます。「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル難読化ユーティリティを使用して、設定ファイルの内容を難読化できます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |
| **-d** *statistic-definition-string* | 統計名と、統計をモニターするスコープ (データベースサーバー、データベース、または接続) を指定します。<br><br>モニター可能な統計の完全なリストについては、「パフォーマンスモニターの統計値」『SQL Anywhere サーバー SQL の使用法』を参照してください。 |
| **-e** *server-definition-string* | モニターするデータベースサーバー名のカンマ区切りのリストを指定します。このオプションを指定しないと、デフォルトのデータベースサーバーに関する統計のみが dbstats ユーティリティから返されます。 |
| **-l** | モニターに使用できる統計がリストされます。 |
| **-o** *filename* | 指定したファイルに、出力メッセージを書き込みます。 |
| **-v** *verbosity* | 指定のレベルでメッセージをレポートします。サポートされる値は、**error**、**warning**、**info** です。デフォルトのレベルは info です。 |
| *interval* | 統計値をレポートする頻度を秒単位で指定します。*interval* を指定すると、統計のレポートを停止するまで、統計値が指定の間隔で返されます。 |
| *iterations* | 統計値の合計レポート回数を指定します。*iterations* に値を指定するには、*interval* 値を指定する必要があります。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## 備考

dbstats ユーティリティでは、共有メモリ接続を使用して、同じコンピューターで稼働している SQL Anywhere サーバーがモニターされます。このユーティリティは、リモートデータベースサーバーのモニターには使用できません。

*statistic-definition-string* は、次の形式で指定します。

[ [ *statistic-name* ][ ,[ *statistic-name* ],[ ... ] ] ] [ :[ *database-servers* ]/[ *databases* ]/[ *connections* ] ]

*database-servers*、*databases*、*connections* の値は、モニターするデータベースサーバー名、データベース名、接続名のカンマ区切りのリストです。名前のカンマ区切りリストを指定するほかに、アスタリスク (*) を使用し、すべてのデータベースサーバー、データベース、接続のモニターを指定することもできます。サポートされているスコープのいずれかの値を指定しないと、そのスコープはモニターされません。

dbstats ユーティリティでオプションをまったく指定しないと、有効なすべてのスコープについて、統計値 ActiveReq、ConnCount、CurrentCacheSize、DiskRead、DiskWrite、MainHeapPages、UnschReq が自動的にモニターされます。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

## 参照

- 「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』
- 「パフォーマンスモニターの統計値」『SQL Anywhere サーバー SQL の使用法』
- 「接続、データベース、データベースサーバーのプロパティ」713 ページ

## 例

デフォルトのデータベースサーバーで、すべてのデータベースと接続について ActiveReq statistic の統計値のみを問い合わせてから、終了します。

```
dbstats -c ActiveReq
```

デフォルトのデータベースサーバーで、すべてのデータベースと接続について ActiveReq statistic の統計値のみを 1 秒に 1 回ずつ無制限に問い合わせます。

```
dbstats -c ActiveReq 1
```

デフォルトのデータベースサーバーで、すべてのデータベースと接続について ActiveReq statistic の統計値のみを 1 秒に 1 回ずつ 5 回だけ問い合わせから、終了します。

```
dbstats -c ActiveReq 1 5
```

すべてのデータベースと接続について ActiveReq と ConnCount の統計値を問い合わせてから、終了します。

```
dbstats -c "[ ActiveReq, ConnCount ]"
```

データベースサーバーのスコープで ActiveReq と ConnCount の統計値を問い合わせてから、終了します。

```
dbstats -c "[ActiveReq, ConnCount]:*"
```

デフォルトのデータベースサーバーで稼働中のデータベース db1 と db2 のみを対象として、データベースのスコープで ActiveReq と ConnCount の統計値を問い合わせてから、終了します。

```
dbstats -c "[ActiveReq, ConnCount]:/db1,db2/"
```

# Ping ユーティリティ (dbping)

データベースサーバーへの接続をテストし、オプションでデータベースへの接続をテストします。

### 構文

**dbping** [ *options* ]

| オプション | 説明 |
|---|---|
| *@data* | 指定された環境変数または設定ファイルからオプションを読み込みます。「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル非表示ユーティリティを使用して、設定ファイルの内容を読みにくくすることができます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |
| **-c** *"keyword=value; ..."* | サーバーと任意でデータベースへの接続に使用された接続パラメーターを指定します。デフォルトでは、dbping は実行中のサーバーに接続します。データベースに接続するためには -d オプションを指定します。「データベースサーバーを見つけられるかどうかのテスト (dbping)」93 ページを参照してください。 |
| **-d** | データベースへの接続をテストします。必要があれば、dbping はサーバーとデータベースを自動的に起動できます。-d があると、dbping は、接続を確立する通常のクライアントアプリケーションのように動作します。<br><br>-d オプションを指定した場合、dbping はサーバーとデータベースに接続する処理の成功をレポートします。-d オプションを指定しない場合、dbping はサーバーを検出すると、処理の成功をレポートします。<br><br>たとえば、サンプルデータベースを実行する blair という名前のデータベースサーバーがある場合、dbping がサーバー blair に接続できるので、次のコマンドは成功します。<br><br>`dbping -c "Server=blair;DBN=other"`<br><br>次のコマンドは失敗し、other データベースが実行中ではなかったので、「**データベースへの ping が失敗しました -- 指定されたデータベースが見つかりません。**」というメッセージが表示されます。<br><br>`dbping -c "Server=blair;DBN=other" -d` |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
|---|---|
| **-en** | 指定したプロパティによって NULL が返された場合に、dbping が失敗を示すリターンコードを返して終了することを指定します。デフォルトでは、 -pc、-pd、-ps のいずれかで指定されたプロパティの値が不明である場合、dbping は NULL を出力し、成功を示すリターンコードを返して終了します。このオプションは、-pc、-pd、-ps を指定する場合のみ使用できます。 |
| **-l** *library* | 使用する ODBC ライブラリを指定します (ファイル拡張子は付けません)。このオプションを使用すると、ODBC ドライバーマネージャーの使用が回避されるので、UNIX オペレーティングシステムで便利です。<br><br>たとえば、次のコマンドは、ODBC ドライバーを直接ロードします。<br><br>dbping -m -c "DSN=SQL Anywhere 12 Demo" -l dbodbc12<br><br>UNIX でスレッド接続ライブラリを使用する場合は、ping ユーティリティのスレッドバージョンである dbping_r を使用します。 |
| **-m** | ODBC を使用して接続を確立します。デフォルトでは、このユーティリティは、Embedded SQL インターフェイスを使用して接続します。 |
| **-o** *filename* | 指定したファイルに、出力メッセージを書き込みます。 |
| **-pc** *property,...* | 指定されている接続プロパティを表示します。プロパティは、カンマで区切って指定します。このオプションを使用するには、データベース接続を確立するために必要な接続情報を指定してください。「接続プロパティ値のアクセス」713 ページを参照してください。<br><br>たとえば、次のコマンドは、接続プロパティとして使用できる fire_triggers オプションの設定を表示します。<br><br>dbping -c ... -pc fire_triggers |

| オプション | 説明 |
|---|---|
| **-pd** *property*[*@db-name*],... | 指定されているデータベースプロパティを表示します。プロパティは、カンマで区切って指定します。「データベースプロパティ値のアクセス」781 ページを参照してください。<br><br>たとえば、次のコマンドは、データベースで使用されているページサイズを表示します。<br><br>dbping -c ... -pd PageSize<br><br>必要に応じて、値を取得するデータベースの名前を指定できます。プロパティに *@db-name* を追加してデータベース名が指定されていない場合は、前のプロパティに指定されたデータベース名が使用されます。<br><br>次のコマンドは、データベース mydb で使用されているページサイズと照合を表示します。<br><br>dbping -c ... -pd PageSize@mydb,Collation |
| **-ps** *property*,... | 指定されているデータベースサーバープロパティを表示します。プロパティは、カンマで区切って指定します。このオプションを使用するには、データベース接続を確立するために必要な接続情報を指定してください。「データベースサーバープロパティ値のアクセス」753 ページを参照してください。<br><br>たとえば、次のコマンドは、データベースサーバーに対するラインセスシート数またはプロセッサー数を表示します。<br><br>dbping -c ... -ps LicenseCount |
| **-q** | クワイエットモードで実行します (メッセージを表示しません)。 |
| **-s** | dbping を実行しているコンピューターとデータベースサーバーを実行しているコンピューター間のネットワークのパフォーマンス情報を返します。接続速度、遅延時間、スループットの概算値が表示されます。通常は、-c オプションを使用して、サーバー上のデータベースに接続するための接続パラメーターを指定します。dbping -s は Embedded SQL 接続でのみ使用できます。-m または -l を指定した場合、-s オプションは無視されます。デフォルトでは、dbping -s は測定する統計ごとに最低 1 秒間、要求のループ処理を実行します。リソースが浪費されることを防ぐために、処理に要する時間にかかわらず、接続と切断は最高 200 回までしか実行されません。低速のネットワークでは、各統計について最低回数の反復処理を実行するために数秒かかることがあります。パフォーマンス統計は概算値であり、クライアントコンピューターとサーバーコンピューターの両方がアイドル状態である方が統計値の精度は高くなります。「Embedded SQL とネットワーク接続パフォーマンスのテスト (dbping)」93 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
|---|---|
| **-st** *time* | -s オプションと同じ情報が返されます。ただし、-st オプションでは、dbping が測定する各統計について要求のループ処理を実行する時間が秒単位で指定されます。このオプションでは、-s を使用した場合より精度の高いタイミング情報を取得できます。「Embedded SQL とネットワーク接続パフォーマンスのテスト (dbping)」93 ページを参照してください。 |
| **-z** | 接続するために使用されたネットワーク通信プロトコルと、他の診断メッセージを表示します。このオプションは、Embedded SQL 接続が行われるときのみ使用できます。つまり、このオプションは、-m または -l と組み合わせて使用することはできません。 |

### 備考

dbping ユーティリティは、接続の問題をデバッグするのに役立つツールです。これは、完全な接続文字列か部分的な接続文字列を取り、サーバーの検索、またはデータベースへの接続が成功したかどうかを示すメッセージを返します。

このユーティリティは、Embedded SQL または ODBC 接続に対して使用できます。jConnect (TDS) 接続に対しては使用できません。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

### 参照

- 「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』
- 「Embedded SQL とネットワーク接続パフォーマンスのテスト (dbping)」93 ページ
- 「データベースサーバーを見つけられるかどうかのテスト (dbping)」93 ページ

## スクリプト実行ユーティリティ (dbrunsql)

Windows Mobile 上で実行されているデータベースに対して、SQL 文やスクリプトファイルを実行できます。

### 構文

**dbrunsql** [ *options* ] [ *SQL-script-file* | *SQL-statement* ]

| オプション | 説明 |
|---|---|
| **-c** *"keyword=value; ..."* | 接続パラメーターを指定します。「接続パラメーター」320 ページを参照してください。 |

| オプション | 説明 |
| --- | --- |
| **-d** | 結果セットからエクスポートされたデータを出力ファイルに書き込みます。-d を指定しない場合は、dbrunsql のすべての出力が出力ファイルに書き込まれます。 |
| **-e** [ **c** \| **p** \| **s** ] | 文の実行中にエラーが発生した場合の動作を制御します。デフォルトでは、ユーザーにエラーが発生したことを知らせるプロンプトを表示します。<br>● **c**　エラーを無視し、文の実行を続行します。<br>● **p**　ユーザーに続行するかどうかを確認するプロンプトを表示します。<br>● **s**　文の実行を停止します。 |
| **-f** [ **f** \| **a** ] | 結果セットをエクスポートするときに dbrunsql が使用するデータフォーマットを指定します。次のいずれかの値を指定できます。<br>● **a**　データのエクスポートに ASCII フォーマットを使用します。<br>● **f**　データのエクスポートに FIXED フォーマットを使用します。これがデフォルトフォーマットです。 |
| **-g** [ + \| - ] | GUI を表示しません。デフォルトでは、dbrunsql GUI が表示されます。 |
| **-o** *filename* | 指定したファイルに、出力メッセージを書き込みます。 |
| **-q** | 出力メッセージを表示しません。これは、文またはスクリプトファイルを使って Interactive SQL を起動したときにのみ便利です。このオプションを指定した場合、エラーメッセージは表示されますが、次の情報は表示されません。<br>● 警告およびその他の致命的でないメッセージ<br>● 結果セットの出力 |
| **-qc** | 文またはスクリプトファイルの実行が完了した後に dbrunsql ウィンドウを閉じます。 |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| **-s** *number* | 結果セットを FIXED フォーマットでエクスポートする場合に、1 つのカラムからフェッチする最大バイト数を指定します。デフォルト値は 255 です。 |
| **-v** | 各 SQL 文のすべての行を dbrunsql の出力に含めます。このオプションを指定せずにスクリプトファイルを実行した場合は、現在実行中の行番号が表示されます。 |

### 備考

dbrunsql ユーティリティでは、データベースに対して SQL 文を実行したりスクリプトファイルを実行したりすることができます。SQL Anywhere スクリプト実行ユーティリティ (dbrunsql) は Windows Mobile でのみサポートされます。

## サーバー列挙ユーティリティ (dblocate)

TCP/IP ネットワーク上のデータベースサーバーを検索します。

### 構文

**dblocate** [ *options* ] [ *host* ]

| オプション | 説明 |
| --- | --- |
| *@data* | 指定された環境変数または設定ファイルからオプションを読み込みます。「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル非表示ユーティリティを使用して、設定ファイルの内容を読みにくくすることができます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |

| オプション | 説明 |
| --- | --- |
| **-d** | 検出したサーバーごとに、サーバー名とアドレスをリストし、その後に各サーバーで実行されているデータベースをカンマで区切ってリストします。リストが 160 文字を超える場合は、トランケートされ、末尾に省略記号 (...) が表示されます。<br><br>SQL Anywhere 9.0.2 以前のデータベースサーバーで実行されているデータベースや、起動時に -dh データベースオプションが指定されたデータベースは、リストされません。「-dh dbeng12/dbsrv12 データベースオプション」309 ページを参照してください。 |
| **-dn** *database-name* | 指定された名前のデータベースを実行しているサーバーのサーバー名とアドレスをリストします。リストが 160 文字を超える場合は、トランケートされ、末尾に省略記号 (...) が表示されます。<br><br>SQL Anywhere 9.0.2 以前のデータベースサーバーで実行されているデータベースや、起動時に -dh データベースオプションが指定されたデータベースは、リストされません。「-dh dbeng12/dbsrv12 データベースオプション」309 ページを参照してください。 |
| **-dv** | 検出したサーバーごとに、サーバー名とアドレスを表示し、さらに各サーバーで実行されているデータベースを別の行にリストします。このリストはトランケートされないため、このオプションを使用すると、-d オプションではトランケートされた部分を確認することができます。<br><br>SQL Anywhere 9.0.2 以前のデータベースサーバーで実行されているデータベースや、起動時に -dh データベースオプションが指定されたデータベースは、リストされません。「-dh dbeng12/dbsrv12 データベースオプション」309 ページを参照してください。 |
| **-n** | コンピューター名ではなく IP アドレスを出力にリストします。コンピューター名の検索処理は低速であるため、パフォーマンスの向上につながることがあります。 |
| **-o** *filename* | 指定したファイルに、出力メッセージを書き込みます。 |

| オプション | 説明 |
| --- | --- |
| **-p** *port-number* | 指定された TCP/IP ポート番号を使用しているサーバーについてのみ、サーバー名とアドレスを表示します。指定する TCP/IP ポート番号は 1 ～ 65535 の範囲内にあることが必要です。 |
| **-q** | クワイエットモードで実行します (メッセージを表示しません)。 |
| **-s** *name* | 指定された名前のサーバーについてのみ、サーバー名とアドレスを表示します。このオプションを使用する場合、-ss オプションは使用できません (両方のオプションを使用すると、サーバーがまったく検出されない可能性があります)。 |
| **-ss** *substr* | 指定された部分文字列が名前に含まれるサーバーについてのみ、サーバー名とアドレスを表示します。このオプションを使用する場合、-s オプションは使用できません (両方のオプションを使用すると、サーバーがまったく検出されない可能性があります)。 |
| **-v** | 完全なサーバー名を表示します。デフォルトでは、dblocate の出力において、40 バイトを超えるデータベースサーバー名はトランケートされます。<br><br>dblocate を含め、バージョン 9.0.2 以前のクライアントは、名前が 40 バイトを超えるバージョン 10.0.0 以降のデータベースサーバーに接続できません。 |

| オプション | 説明 |
| --- | --- |
| *host* | 指定された IP アドレスまたはホスト名を持つコンピューターで実行されているデータベースサーバーのみをリストします。たとえば、次のコマンドは、コンピューター jfrancis 上のサーバーを検索します。<br><br>dblocate jfrancis<br><br>-n を指定するかどうかにかかわらず、ホスト名や IP アドレスはどのような形式でもかまいません。たとえば、IP アドレスが 1.2.3.4 であるコンピューター myhost.mycompany.com で実行されているサーバーがあるとします。このコンピューター上で実行されているサーバーだけを、mycompany.com ドメインのコンピューターからリストする場合は、dblocate myhost、dblocate myhost.mycompany.com、dblocate 1.2.3.4 のいずれも使用できます。 |

## 備考

サーバー列挙ユーティリティ (dblocate) は、直接接続されているネットワークを介して TCP/IP 上で実行されている SQL Anywhere データベースサーバーを検索し、データベースサーバーとそのアドレスのリストを出力します。このリストには代替サーバー名も含まれています。「-sn dbsrv12 オプション」315 ページを参照してください。

ネットワークによっては、dblocate が結果を出力するまでに数秒かかることがあります。

**注意**
Mac OS X で 2638 以外の TCP/IP ポートを使用しているデータベースサーバーは、-p オプションでその TCP/IP ポートを指定しても、dblocate では検出されません。「ServerPort (PORT) プロトコルオプション」396 ページを参照してください。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

データベースサーバーは、サーバー自体を LDAP サーバーとして登録でき、企業内のすべてのサーバーを追跡することができます。これにより、クライアントと dblocate は、サーバーが WAN または LAN にあるかどうかにかかわらず、IP アドレスを指定せずにファイアウォールを介してサーバーを検索できます。LDAP は TCP/IP でネットワークサーバー上でのみ使用されます。

同じサーバー名が複数回検索された場合、dblocate は -n オプションが指定されていなくても、各ホストの IP アドレスを表示します。同じサーバー名が見つかるのは、複数の IP アドレスを持つコンピューター (たとえば、コンピューターに複数のネットワークカードがある) でサーバーを実行している場合、またはネットワークサーバーをリモートコンピューターで実行し、同じ名前を持つパーソナルサーバーをローカルコンピューターで実行している場合です。

-sb 0 で起動したデータベースサーバーは dblocate では見つかりません。

### 参照

- 「LDAP サーバーを使用した接続」103 ページ
- 「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』
- 「データベースサーバーの検出」80 ページ
- 「db_string_ping_server 関数」『SQL Anywhere サーバー プログラミング』

## サーバーライセンス取得ユーティリティ (dblic)

SQL Anywhere データベースサーバーまたは Mobile Link サーバーに、ソフトウェアライセンスを適用します。

### 構文

**dblic** [ *options* ] *license-file* [ "*user-name*" "*company-name*" ]

| オプション | 説明 |
|---|---|
| *@data* | 指定された環境変数または設定ファイルからオプションを読み込みます。「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル非表示ユーティリティを使用して、設定ファイルの内容を読みにくくすることができます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |
| **-k** *registration-key* | SQL Anywhere のライセンスキーを指定します。このオプションを使用して、データベースサーバーのエディションを変更できます。たとえば、手元にある開発版をワークグループ版に変更する場合は、iAnywhere から入手したワークグループ版の登録キーを指定できます。<br><br>このオプションは、-l オプションとは一緒に使用できません。 |

| オプション | 説明 |
| --- | --- |
| **-l** *type* | ソフトウェアのライセンス契約に記述されているライセンスモデルと一致するライセンスタイプを指定します。サポートされるライセンスタイプは次のとおりです。<br><br>● **Perseat**　パーシートライセンスは、データベースサーバーに対するクライアント接続の数を制限します。パーシートライセンスでは、-gt オプション、または実行しているエディションによる制限がネットワークデータベースサーバーにある場合を除き、コンピューターにあるすべての CPU がネットワークデータベースサーバーで使用されます。パーソナルサーバーは CPU が 1 つに制限されています。<br><br>● **Processor**　プロセッサーライセンスは、データベースサーバーが使用できる独立した物理プロセッサーの数を制限します。データベースサーバーで使用できる CPU 数は、-gt オプション、または実行している SQL Anywhere のエディションによってさらに制限されている場合があります。パーソナルデータベースサーバーは CPU が 1 つに制限されています。<br><br>このライセンスタイプでは、データベースサーバーは、各物理プロセッサーを 1 つの CPU として扱います。デュアルコアまたはハイパースレッドのプロセッサーを複数のプロセッサーとして扱うことはありません。プロセッサーライセンスを取得している場合、データベースサーバーに対するクライアント接続の数に制限はありません。<br><br>このオプションは、-k オプションとは一緒に使用できません。 |
| **-o** *filename* | 指定したファイルに、出力メッセージを書き込みます。 |
| **-q** | クワイエットモードで実行します (メッセージを表示しません)。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| **-u** *license-number* | ライセンスされるユーザーまたはプロセッサーの総数を指定します。ライセンスを追加する場合でも、追加ライセンスの数ではなく、総数を指定します。 |
| *license-file* | ライセンスされているパーソナルデータベースサーバー、ネットワークデータベースサーバー、または Mobile Link サーバーのサーバー実行プログラムまたはライセンスファイルのパスと名前を指定します。<br><br>ライセンスファイル名だけを入力することで、サーバー実行プログラムの現在のライセンス情報を表示できます。 |
| *user-name* | ライセンスのユーザー名を指定します。この名前は、起動時にデータベースサーバーメッセージウィンドウに表示されます。名前にスペースが含まれている場合、二重引用符で囲んでください。 |
| *company-name* | ライセンスの会社名を指定します。この名前は、起動時にデータベースサーバーメッセージウィンドウに表示されます。名前にスペースが含まれている場合、二重引用符で囲んでください。 |

**備考**

サーバーライセンス取得ユーティリティでは、SQL Anywhere データベースサーバーまたは Mobile Link サーバーにライセンスユーザーまたはライセンスプロセッサーを追加できます。このユーティリティは、ライセンス契約に基づいて、許可されたライセンスユーザー数またはライセンスプロセッサー数の範囲内で使用してください。このコマンドの実行によって、ライセンスが付与されることはありません。データベースサーバーで使用できる CPU 数には、SQL Anywhere のエディションまたは -gt サーバーオプションも影響する場合があります。次の項を参照してください。


- 「エディションとライセンス」『SQL Anywhere 12 紹介』
- 「-gt dbeng12/dbsrv12 サーバーオプション」244 ページ


このユーティリティでは、パーソナルデータベースサーバー、ネットワークデータベースサーバー、または Mobile Link サーバーによって起動時に表示されるユーザー名や会社名を変更することもできます。

また、ライセンスファイル名だけを入力することで、パーソナルデータベースサーバーやネットワークデータベースサーバーを起動しないで、現在のライセンス情報を表示することも可能です。

ライセンス情報は、サーバーの実行プログラムと同じディレクトリにある *.lic* ファイルに格納されます。サーバーは、実行中の実行プログラムと同じベースファイル名を持つ *.lic* ファイルを探します。たとえば、データベースサーバーの実行プログラムが *myserver.exe* という名前の場合、サーバーは *myserver.lic* という名前のライセンスファイルを探します。デフォルトでは、次の名前が使用されます。

| 実行プログラム | ライセンスファイル名 |
|---|---|
| SQL Anywhere パーソナルデータベースサーバー (dbeng12) | *dbeng12.lic* |
| SQL Anywhere ネットワークデータベースサーバー (dbsrv12) | *dbsrv12.lic* |
| Mobile Link サーバー (mlsrv12) | *mlsrv12.lic* |

サーバーを起動するときに、対応する *.lic* ファイルが使用できない場合は、サーバーは起動されません。ライセンスファイルは、SQL Anywhere インストールプログラムによって作成されます。dblic ユーティリティは、既存のライセンスを修正するだけで、新しいライセンスファイルを作成することはありません。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

UNIX では、データベースサーバーの実行プログラムはデフォルトで書き込みができないので、サーバーライセンス取得 (dblic) ユーティリティを使用すると失敗します。実行プログラムは、サーバーライセンス取得ユーティリティを使用する前に (chmod +w を使用するなどして) 書き込み可能にしてください。

キャッシュされた接続は、パーシートライセンスとしてカウントされます。

SQL Anywhere ライセンスの詳細については、http://www.ianywhere.jp/sas/1056242-sa_licensing.html を参照してください。

## 参照

- 「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』

## 例

次のコマンドをデータベースサーバーの実行プログラムと同じディレクトリで実行すると、ユーザー名 Sys Admin、会社名 My Co という設定で、50 ユーザーのライセンスが、Microsoft Windows ネットワークデータベースサーバーに適用されます。コマンドは、1 行に入力してください。

```
dblic -l perseat -u 50 "c:¥Program Files¥SQL Anywhere 12¥Bin32¥dbsrv12.lic" "Sys Admin" "My Co"
```

ライセンスの適用が成功すると、画面に次のメッセージが表示されます。

```
SQL Anywhere サーバーライセンス取得ユーティリティバージョン 12.0.1.3149
ライセンスが正常に適用されました。
評価版
アドオン : 高可用性
アドオン : インメモリモード
アドオン : スケールアウトノード
```

```
ライセンスされるノード : 50
ユーザー : Sys Admin
会社名 : My Co
インストールキー :
```

次のコマンドは、データベースサーバーのライセンスについての情報を返します。

```
dblic "c:¥Program Files¥SQL Anywhere 12¥Bin32¥dbsrv12.lic"
```

# Linux 用サービスユーティリティ (dbsvc)

SQL Anywhere サービスの作成、変更、削除を行います。

### 構文

**dbsvc** [ *modifier-options* ] **-d** *svc*

**dbsvc** [ *modifier-options* ] **-g** *svc*

**dbsvc** [ *modifier-options* ] **-l**

**dbsvc** [ *modifier-options* ] **-status** *svc*

**dbsvc** [ *modifier-options* ] **-u** *svc*

**dbsvc** [ *modifier-options* ] *creation-options* **-w** *svc* [ *options* ]

**dbsvc** [ *modifier-options* ] **-x** *svc*

| 主要オプション | 説明 |
|---|---|
| **-d** *service-name* | サービスリストから指定のサービスを削除します。-y を指定すると、確認メッセージを表示せずにサービスを削除します。 |
| **-g** *service-name* | サービスの定義を表示します。 |
| **-l** | 使用できる SQL Anywhere サービスをリストします。 |
| **-u** *service-name* | *service-name* という名前のサービスを起動します。 |

| 主要オプション | 説明 |
| --- | --- |
| **-w** *executable parameters* | 新しいサービスを作成するか、同名のサービスが存在する場合はそれを上書きします。-y を指定すると、確認メッセージを表示せずに既存のサービスを上書きします。実行中のサービスは削除できません。<br><br>作成するサービスに適したパラメーターを指定します。<br><br>次の項を参照してください。<br><br>● **dbsrv12 と dbeng12** 「SQL Anywhere データベースサーバーの構文」<br><br>● **mlsrv12** 「Mobile Link サーバーオプション」『Mobile Link サーバー管理』<br><br>● **dbmlsync** 「dbmlsync 構文」『Mobile Link クライアント管理』<br><br>● **dbremote** 「SQL Remote Message Agent ユーティリティ (dbremote)」『SQL Remote』<br><br>● **dbns12** 「Broadcast Repeater ユーティリティ (dbns12)」<br><br>● **rshost** 「Relay Server ステイトマネージャー」『Relay Server』<br><br>● **rsoe** 「Outbound Enabler」『Relay Server』 |
| **-x** *service-name* | *service-name* という名前のサービスを停止します。 |

| 作成オプション | 説明 |
| --- | --- |
| **-a** *acct* | アカウントを指定します。すべてのサービスは、Linux アカウントの下で実行されます。独自に作成したアカウントで実行する場合は、-a オプションでアカウントを指定する必要があります。 |

| 作成オプション | 説明 |
| --- | --- |
| **-as** | Linux デーモンアカウントの下でサービスを実行します。すべてのサービスは、Linux アカウントの下で実行されます。パスワードは必要ありません。-a または -as のいずれかを必ず使用してください。 |
| **-od** | システム情報ファイルのロケーションを指定します (必要な場合)。 |
| **-pr** | Linux プロセスのナイスレベルを設定します。 |
| **-rl** | サービスを起動するランレベルを指定します。 |
| **-rs** *service-name* | dbsvc によって生成される init スクリプトの Required-Start ヘッダーで、サービスのリストにサービスを追加します。このヘッダーは、オペレーティングシステムでサービスの開始順序を決めるのに役立ちます。<br><br>指定されたサービス名は、すでに存在するかどうかが検証されます。サービス名は、**SA_***service-name* の形式で指定します。 |
| **-s** | SQL Anywhere サービスの起動時の動作を設定します。起動時の動作は Automatic または Manual に設定できます。デフォルトは Manual です。 |
| **-status** | サービスの実行ステータスを返します。 |

| 作成オプション | 説明 |
| --- | --- |
| **-t** *type* | このサービスのタイプを指定します。次のタイプから選択できます (代替名はカッコ内に示します)。<br><br>● **Network**　SQL Anywhere ネットワークデータベースサーバー (dbsrv12) のサービスを作成します。「SQL Anywhere データベースサーバーの構文」189 ページを参照してください。<br><br>● **Personal (Standalone)**　これがデフォルトです。SQL Anywhere パーソナルデータベースサーバー (dbeng12) のサービスを作成します。「SQL Anywhere データベースサーバーの構文」189 ページを参照してください。<br><br>● **DBRemote**　SQL Remote Message Agent のサービスを作成します。「SQL Remote Message Agent ユーティリティ (dbremote)」『SQL Remote』を参照してください。<br><br>● **Mobile Link**　Mobile Link サーバー (mlsrv12) のサービスを作成します。「mlsrv12 構文」『Mobile Link サーバー管理』を参照してください。<br><br>● **DBMLSync (MLSync)**　Mobile Link 同期クライアント (dbmlsync) のサービスを作成します。「dbmlsync 構文」『Mobile Link クライアント管理』を参照してください。<br><br>● **DBNS**　Broadcast Repeater ユーティリティ (dbns12) のサービスを作成します。「Broadcast Repeater ユーティリティ (dbns12)」921 ページを参照してください。<br><br>● **RSHOST**　Relay Server ステイトマネージャーのサービスを作成します。「Relay Server ステイトマネージャーのコマンドラインの構文」『Relay Server』を参照してください。<br><br>● **RSOE**　Outbound Enabler のサービスを作成します。「Outbound Enabler の構文」『Relay Server』を参照してください。 |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 変更オプション | 説明 |
| --- | --- |
| **-cm** | サービスの作成に使用するコマンドを表示します。このオプションを使用すると、作成コマンドをファイルに出力できます。このファイルを使用して、別のコンピューターにサービスを追加したり、変更が加えられたサービスを元の状態にリストアしたりすることができます。-cm とともに -g オプションまたは -l オプションを指定しないとコマンドは失敗します。-g を指定すると、指定のサービスの作成コマンドが表示されるのに対し、-l を指定するとすべてのサービスの作成コマンドが表示されます。 |
| **-q** | データベースサーバーメッセージウィンドウにメッセージを表示しないようにします。既存のサービスの変更時または削除時にこのオプションを指定する場合、-y も指定しないと操作は失敗します。 |
| **-y** | 確認メッセージを表示することなく、処理を実行します。このオプションは、-w または -d オプションと一緒に使用できます。既存のサービスの変更時または削除時に -q を指定する場合、-y も指定しないと操作は失敗します。 |

**備考**

サービスは、一連のオプションを使ってデータベースサーバーやその他のアプリケーションを実行します。このユーティリティを使用して、Linux 上で動作している SQL Anywhere サービスを包括的な方法で管理できます。

通常、サービスの作成環境と実行環境は異なるため、サービスの作成時にはデータベースファイルに完全に修飾された名前を付けることをおすすめします。また、データソース名にはスペースを入れないでください。

サービスを作成、編集、削除するには、*/etc/init.d* ディレクトリに対するパーミッションが必要です。

SQL Anywhere サービスが Linux で実行されている場合、PID ファイルは */var/run* ディレクトリで作成されます。このファイルには、dbsvc プロセスの PID が含まれます。ファイル名は *SA_service-name.pid* になります。このファイルを他の Linux ツールで使用して、プロセスを検索したり、サービスをモニターしたりできます。

dbsvc ユーティリティでは、ほとんどの Linux サービスと同じように、サービスファイルが */etc/init.d* に作成されます。サービス名は、**SA_***service-name* の形式で指定します。たとえば、myserv

という名前のサービスを作成した場合、このサービスを起動するには、次のようなコマンドを発行します。

```
/etc/init.d/SA_myserv start
```

次のコマンドは、サービスのステータスを返します。

```
/etc/init.d/SA_myserv status
```

次のコマンドは、サービスの使用状況に関する情報を返します。

```
/etc/init.d/SA_myserv
```

**注意**
dbsvc コマンドにシェル固有の文字 (セミコロンまたはカッコ) が含まれている場合は、サービスに対するパラメーター用に設定ファイルを使用することをおすすめします。「設定ファイル」909 ページを参照してください。

**例**

指定のサーバーを指定のパラメーターで起動する、myserv という名前のパーソナルサーバーサービスを作成します。このサーバーは LocalSystem ユーザーとして実行されます。

```
dbsvc -as -w myserv -n myeng -c 8m "/tmp/demo.db"
```

mynetworkserv という名前のネットワークサーバーサービスを作成します。このサーバーはローカルアカウントで実行され、コンピューターを再起動すると自動的に起動します。

```
dbsvc -as -t network -w mynetworkserv -x tcpip -c 8m "/tmp/demo.db"
```

サービス myserv についての詳細をすべてリストします。

```
dbsvc -g myserv
```

myserv という名前のサービスを、確認メッセージを表示せずに削除します。

```
dbsvc -y -d myserv
```

mysyncservice という名前のサービスを作成します。

```
dbsvc -as -t dbmlsync -w mysyncservice -c "/tmp/CustDB.db" -o syncinfo.txt
```

service_1 サービスを作成するコマンドを生成し、そのコマンドをコンソールに出力します。

```
dbsvc -cm -g service_1
```

コンソールには次のように表示されます。

```
'dbsvc -t Standalone -as -y -w "service_1" -n'
```

dbsvc を使用してサービスを起動します。

```
dbsvc -u myserv
```

dbsvc を使用してサービスを停止します。

```
dbsvc -x myserv
```

dbsvc を使用してサービスを取得します。

```
dbsvc -status myserv
```

# Windows 用サービスユーティリティ (dbsvc)

SQL Anywhere サービスの作成、変更、削除を行います。

### 構文

**dbsvc** [ *modifier-options* ] **-d** *svc*

**dbsvc** [ *modifier-options* ] **-g** *svc*

**dbsvc** [ *modifier-options* ] **-l**

**dbsvc** [ *modifier-options* ] **-u** *svc*

**dbsvc** [ *modifier-options* ] *creation-options* **-w** *svc details*

**dbsvc** [ *modifier-options* ] **-x** *svc*

*details*:

<*full-executable-path*> [ *options* ]

| 主要オプション | 説明 |
|---|---|
| *@data* | 指定された環境変数または設定ファイルからオプションを読み込みます。「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル非表示ユーティリティを使用して、設定ファイルの内容を読みにくくすることができます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |
| **-d** *service-name* | サービスリストから指定のサービスを削除します。-y を指定すると、確認メッセージを表示せずにサービスを削除します。 |
| **-g** *service-name* | パスワード以外のサービスの定義をリストします。 |
| **-l** | 使用できる SQL Anywhere サービスをリストします。 |

| 主要オプション | 説明 |
| --- | --- |
| **-u** *service-name* | *service-name* という名前のサービスを起動します。 |

| 主要オプション | 説明 |
| --- | --- |
| **-w** *executable parameters* | 新しいサービスを作成するか、同名のサービスが存在する場合はそれを上書きします。-y を指定すると、確認メッセージを表示せずに既存のサービスを上書きします。実行中のサービスは削除できません。<br><br>サービスとして使用する実行プログラムのフルパスを指定します。これは、その下でサービスが実行されるアカウントのパスの中に適切な SQL Anywhere インストールディレクトリがない場合があるからです。<br><br>作成するサービスに適したパラメーターを指定します。<br><br>次の項を参照してください。<br><br>● **dbsrv12 と dbeng12** 「SQL Anywhere データベースサーバーの構文」 189 ページ<br><br>● **mlsrv12** 「Mobile Link サーバーオプション」『Mobile Link サーバー管理』<br><br>● **dbmlsync** 「dbmlsync 構文」『Mobile Link クライアント管理』<br><br>● **dblsn** 「Windows デバイス用の Mobile Link Listener ユーティリティ (dblsn)」『Mobile Link サーバー起動同期』<br><br>● **dbremote** 「SQL Remote Message Agent ユーティリティ (dbremote)」『SQL Remote』<br><br>● **dbns** 「Broadcast Repeater ユーティリティ (dbns12)」 921 ページ<br><br>● **rshost** 「サービスとしての Relay Server ステイトマネージャー」『Relay Server』<br><br>● **rsoe** 「サービスとしての Outbound Enabler」『Relay Server』<br><br>● **dbvss12** 「SQL Anywhere ボリュームシャドウコピーサービス (VSS)」 1084 ページ |

| 主要オプション | 説明 |
| --- | --- |
| | ● **mlagent** 「Mobile Link エージェント」『Mobile Link サーバー管理』 |
| **-x** *service-name* | *service-name* という名前のサービスを停止します。 |

| 作成オプション | 説明 |
| --- | --- |
| **-a** *acct* | Microsoft Windows アカウントを指定します。すべてのサービスは、Microsoft Windows アカウントの下で実行されます。作成したアカウントの下で実行する場合は、-a オプションでアカウントを指定し、-p オプションでパスワードを指定します。<br><br>サービスとしてログインする権限は、LocalSystem を除くすべてのアカウントに必要です。アカウントのサービスとしてログインする権限が有効でない場合は、有効にするよう要求されます。-y オプションも指定すると、dbsvc はプロンプトを表示することなくサービスとしてログインする権限を付与しようとします。-y オプションを指定しないで -q オプションだけを指定すると、サービスとしてログインする権限を有効にするよう要求するプロンプトは表示されず、dbsvc は失敗します。 |
| **-as** | Microsoft Windows LocalSystem アカウントの下でサービスを実行します。パスワードは必要ありません。-a または -as のいずれかを必ず使用してください。すべてのサービスは、Microsoft Windows アカウントの下で実行されます。 |
| **-i** | アイコンが表示され、これをダブルクリックするとデータベースサーバーメッセージウィンドウが表示されます。 |
| **-p** | サービスが実行されるときのアカウントのパスワードを指定します。このオプションは、-a オプションと一緒に使用します。 |
| **-rg** *dependency,...* | 作成するサービスが起動できるよう事前に起動しておく必要のある 1 つ以上の起動順序グループを指定します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 作成オプション | 説明 |
| --- | --- |
| **-rs** *dependency,...* | リストに含まれるすべてのサービスが起動してから、作成したサービスの起動を許可するように指定します。<br><br>サービス名の表示名を指定できます。サービス名が検証され、すでに存在するかどうかが確認されます。指定されたサービス名が見つからない場合、表示名が一致するサービスが存在するかどうかがサービスユーティリティ (dbsvc) によってチェックされます。<br><br>接続が正しく開始することを確認するため、TCP/IP 用のトランスポートデータインターフェイス (TDI) で依存性を作成することをおすすめします。TDI の詳細については、http://msdn.microsoft.com/en-us/library/ms797340.aspx を参照してください。 |
| **-s** *startup* | SQL Anywhere サービスの起動時の動作を設定します。起動時の動作は Automatic、Manual、または Disabled に設定できます。デフォルトは Manual です。 |
| **-sd** *description* | サービスの説明を示します。説明は、Windows のサービス マネージャーに表示されます。 |
| **-sn** *name* | サービスの名前を示します。この名前は、Windows のサービス マネージャーに表示されます。-sn オプションを指定しない場合、デフォルトのサービス名は SQL Anywhere - *svc* です。たとえば、次のサービスにはデフォルトで SQL Anywhere - myserv という名前が付けられます。<br><br>dbsvc -as -w myserv<br>"C:¥Program Files¥SQL Anywhere 12¥Bin32¥dbeng12.exe"<br><br>サービス名 myserv が Windows のサービスマネージャーに表示されるようにするには、次のコマンド (全体を 1 行に入力) を実行する必要があります。<br><br>dbsvc -as -sn myserv -w myserv<br>"C:¥Program Files¥SQL Anywhere 12¥Bin32¥dbeng12.exe" |

| 作成オプション | 説明 |
| --- | --- |
| **-t** *type* | このサービスのタイプを指定します。次のタイプから選択できます (代替名はカッコ内に示します)。<br><br>● **Network**　SQL Anywhere ネットワークデータベースサーバー (dbsrv12) のサービスを作成します。「SQL Anywhere データベースサーバーの構文」189 ページを参照してください。<br><br>● **Personal (Standalone)**　これがデフォルトです。SQL Anywhere パーソナルデータベースサーバー (dbeng12) のサービスを作成します。「SQL Anywhere データベースサーバーの構文」189 ページを参照してください。<br><br>● **DBRemote**　SQL Remote Message Agent のサービスを作成します。「SQL Remote Message Agent ユーティリティ (dbremote)」『SQL Remote』を参照してください。<br><br>● **Mobile Link**　Mobile Link サーバー (mlsrv12) のサービスを作成します。「mlsrv12 構文」『Mobile Link サーバー管理』を参照してください。<br><br>● **DBMLSync (MLSync)**　Mobile Link 同期クライアント (dbmlsync) のサービスを作成します。「dbmlsync 構文」『Mobile Link クライアント管理』を参照してください。<br><br>● **DBNS**　Broadcast Repeater ユーティリティ (dbns12) のサービスを作成します。「Broadcast Repeater ユーティリティ (dbns12)」921 ページを参照してください。<br><br>● **DBLSN (LSN)**　Mobile Link Listener ユーティリティのサービスを作成します。「Windows デバイス用の Mobile Link Listener ユーティリティ (dblsn)」『Mobile Link サーバー起動同期』を参照してください。<br><br>● **DBVSS (VSS)**　SQL Anywhere VSS ライターのサービスを作成します。「SQL Anywhere ボリュームシャドウコピーサービス (VSS)」1084 ページを参照してください。<br><br>● **RSHOST**　Relay Server ステイトマネージャーのサービスを作成します。「Relay Server ステイトマネージャーのコマンドラインの構文」『Relay Server』を参照してください。<br><br>● **RSOE**　Outbound Enabler のサービスを作成します。「Outbound Enabler の構文」『Relay Server』を参照してください。<br><br>● **MLAGENT**　Mobile Link エージェントのサービスを作成します。「Mobile Link エージェント」『Mobile Link サーバー管理』を参照してください。 |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 変更オプション | 説明 |
| --- | --- |
| **-cm** | サービスの作成に使用するコマンドを表示します。このオプションを使用すると、作成コマンドをファイルに出力できます。このファイルを使用して、別のコンピューターにサービスを追加したり、変更が加えられたサービスを元の状態にリストアしたりすることができます。-cm とともに -g オプションまたは -l オプションを指定しないとコマンドは失敗します。-g を指定すると、指定のサービスの作成コマンドが表示されるのに対し、-l を指定するとすべてのサービスの作成コマンドが表示されます。<br><br>指定のサービスが存在しない場合は、サービスを削除するコマンドが生成されます。たとえば、コンピューターに service_1 が存在しない場合、dbsvc -cm -g service_1 は service_1 サービスを削除する次のコマンドを返します。<br><br>dbsvc -y -d "service_1"<br><br>サービスが LocalSystem アカウントを使用しない場合は、パスワードを取り出すことはできないため、生成されるコマンドにはパスワードは含まれません。-a user -p password を使用してサービスを作成した場合、出力には -a user のみが含まれます。 |
| **-o** *log-file* | サービスユーティリティ (dbsvc) の出力を指定されたファイルに書き込みます。-o オプションは、-d、-g、-l、-u、-x の前に指定してください。dbsvc とサービスとして実行する実行ファイル (データベースサーバーなど) の両方に指定すると、それぞれについてログファイルが作成されます。次に例を示します。<br><br>dbsvc -o out1.txt -y -as -w mydsn "%SQLANY12%¥bin32¥dbsrv12" -n mysrv -o c:¥out2.txt<br><br>この場合は、dbsvc の出力が *out1.txt* に書き込まれ、データベースサーバーの出力は *c:¥out2.txt* に書き込まれます。 |
| **-q** | データベースサーバーメッセージウィンドウにメッセージを表示しないようにします。-q を指定する場合は、-o オプションを使用してメッセージを書き込むファイルを指定するようおすすめします。既存のサービスの変更時または削除時にこのオプションを指定する場合、-y も指定しないと操作は失敗します。 |
| **-y** | 確認メッセージを表示することなく、処理を実行します。このオプションは、-w または -d オプションと一緒に使用できます。既存のサービスの変更時または削除時に -q を指定する場合、-y も指定しないと操作は失敗します。 |

### 備考

サービスは、一連のオプションを使ってデータベースサーバーやその他のアプリケーションを実行します。このユーティリティでは、Windows 上で動作している SQL Anywhere サービスを包括的な方法で管理できます。サービスユーティリティを使用するには、ローカルコンピューター上で Administrators グループのメンバーである必要があります (dbsvc)。

サービスユーティリティにアクセスするには、次の方法があります。

●Sybase Central の**サービス作成ウィザード**を使用する。

●コマンドプロンプトで、サービスユーティリティを使用する (dbsvc)。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

### 参照


- 「Windows サービスの作成」172 ページ
- 「Windows サービス」171 ページ
- 「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』


### 例

指定のサーバーを指定のパラメーターで起動する、myserv という名前のパーソナルサーバーサービスを作成します。このサーバーは LocalSystem ユーザーとして実行されます。

```
dbsvc -as -w myserv
"C:¥Program Files¥SQL Anywhere 12¥Bin32¥dbeng12.exe"
-n myeng -c 8m "c:¥temp¥mysample.db"
```

mynetworkserv という名前のネットワークサーバーサービスを作成します。このサーバーはローカルアカウントで実行され、コンピューターを再起動すると自動的に起動します。

```
dbsvc -as -s auto -t network -w mynetworkserv
"C:¥Program Files¥SQL Anywhere 12¥Bin32¥dbsrv12.exe"
-x tcpip -c 8m "c:¥temp¥mysample.db"
```

サービス myserv についての詳細をすべてリストします。

```
dbsvc -g myserv
```

myserv という名前のサービスを、確認メッセージを表示せずに削除します。

```
dbsvc -y -d myserv
```

Workstation サービスと TDI グループに依存するサービスを作成します。

```
dbsvc -rs lanmanworkstation -rg TDI -w ...
```

mysyncservice という名前のサービスを作成します。

```
dbsvc -as -s manual -t dbmlsync -w mysyncservice
"C:¥Program Files¥SQL Anywhere 12¥Bin32¥dbmlsync.exe"
-c "SQL Anywhere 12 CustDB"
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

service_1 サービスを作成するためのコマンドを生成し、それを *restoreservice.bat* というファイルに出力します。

```
dbsvc -cm -g service_1 > restoreservice.bat
```

*restoreservice.bat* ファイルには以下が含まれています。

```
dbsvc -t Standalone -s Manual -as -y -w "service_1"
"C:¥Program Files¥SQL Anywhere 12¥Bin32¥dbeng12.exe"
```

手動で起動される Mobile Link Listener サービスを作成します。

```
dbsvc -as -i -w myListener
"C:¥Program Files¥SQL Anywhere 12¥Bin32¥dblsn.exe" "@c:¥temp¥dblsn.opt"
```

myListener サービスを起動します。

```
dbsvc -u myListener
```

myListener サービスを停止します。

```
dbsvc -x myListener
```

データベースサーバーの起動時に自動的に開始するボリュームシャドウコピーサービス (VSS) サービスを作成します。

```
dbsvc -as -s Automatic -t vss -w SAVSSWriter
"C:¥Program Files¥SQL Anywhere 12¥Bin32¥dbvss12.exe"
```

# SQL Anywhere コンソールユーティリティ (dbconsole)

データベースサーバー接続の管理機能とモニタリング機能を提供します。

### 構文

**dbconsole** [ *options* ]

| オプション | 説明 |
|---|---|
| *@data* | 指定された環境変数または設定ファイルからオプションを読み込みます。「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル非表示ユーティリティを使用して、設定ファイルの内容を読みにくくすることができます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |

| オプション | 説明 |
| --- | --- |
| **-c** *"keyword=value; ..."* | 接続パラメーターを指定します。「接続パラメーター」320 ページを参照してください。 |
| **-datasource** *DSN-name* | 接続先の ODBC データソースを指定します。このオプションを使用するために、SQL Anywhere JDBC ドライバーを使用している必要はありません。 |
| **-host** *hostname* | データベースサーバーを実行するコンピューターのホスト名または IP アドレスを指定します。現在のコンピューターを表す localhost という名前を使用できます。 |
| **-port** *port-number* | データベースサーバーが実行されているポート番号を指定します。SQL Anywhere のデフォルトポート番号は **2638** です。 |

### 備考

SQL Anywhere コンソールユーティリティでは、クライアントコンピューターからサーバーをモニターできます。このユーティリティはネットワークサーバーモニターとも呼ばれます。これを使うと、使用しているネットワーク上でデータベースサーバーにログオンしているユーザーを追跡できます。また、ローカルクライアント画面へのデータベースサーバーとクライアント統計の表示、ユーザーの切断、データベースサーバーの設定を行うことができます。SQL Anywhere コンソールは、複数の接続に関する情報を表示できます。

SQL Anywhere コンソールは、Windows Mobile、IBM AIX、HP-UX、HP-UX Itanium を除き、サポートされるすべてのプラットフォームで使用可能です。これらのプラットフォームでは、接続レベル、サーバーレベル、データベースレベルのプロパティを使用して情報を取得したり、SQL Anywhere コンソールをサポートするオペレーティングシステム (Windows、Mac OS X、Linux など) を実行するコンピューターからサーバーをモニターしたりすることができます。

プロパティ値の取得については、「接続、データベース、データベースサーバーのプロパティ」713 ページを参照してください。

### 参照

● 「SQL Anywhere コンソールユーティリティ (dbconsole)」905 ページ

## サーバーバックグラウンド起動ユーティリティ (dbspawn)

バックグラウンドでデータベースサーバーを起動します。

### 構文

**dbspawn** [ *options* ] *server-command*

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| *@data* | 指定された環境変数または設定ファイルからオプションを読み込みます。同じ名前の環境変数と設定ファイルが両方存在する場合は、環境変数が使用されます。<br><br>サーバーバックグラウンド起動ユーティリティ (dbspawn) では、作成するコマンドで設定ファイル参照を指定できますが、dbspawn ユーティリティのオプションを使用して設定ファイルを指定することはできません。たとえば、次の最初のコマンドはサポートされていますが、2 つ目のコマンドはサポート**されていません**。<br><br>dbspawn dbeng12 @myconfig.ini<br><br>dbspawn @spawnopts.ini dbeng12 demo.db<br><br>詳細については、「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル難読化ユーティリティを使用して、設定ファイルの内容を難読化できます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |
| **-f** | デフォルトのデータベースサーバーがすでに存在する場合でも、dbspawn を使用してデータベースサーバーを起動します。 |
| **-p** | データベースサーバーの起動後に返される、データベースサーバープロセスのオペレーティングシステムプロセス ID を指定します。次に例を示します。<br><br>新しいプロセス ID は 306 です |
| **-q** | クワイエットモードで実行します (メッセージを表示しません)。 |
| *server-command* | データベースサーバーを開始するためのコマンドラインを指定します。「SQL Anywhere データベースサーバーの構文」189 ページを参照してください。 |

### 備考

dbspawn ユーティリティは、サーバーをバックグラウンドで起動するために用意されています。dbspawn は、バックグラウンドでサーバーを起動し、終了コード 0 (成功) または 0 以外の値 (失敗) を返します。新しいサーバーが正常に起動された場合は、データベースサーバーの初期処理が完了し、要求を受信できる状態になるまで、dbspawn は終了しません。

デフォルトでは、dbspawn はデフォルトデータベースサーバーが現在実行中のコンピューターでは新しいサーバーを起動できません。デフォルトデータベースサーバーが実行中のコンピューターでサーバーを起動するには、-f オプションを指定します。

終了コードの詳細については、「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』を参照してください。

dbspawn ユーティリティは、バッチファイルからサーバーを起動するのに役立ちます。特に、バッチファイルの後続コマンドが要求を受け入れるサーバーを必要とする場合に便利です。

指定するパスに 1 つ以上のスペースが含まれている場合は、パス全体を二重引用符で囲む必要があります。次に例を示します。

```
dbspawn dbeng12 "c:¥my databases¥mysalesdata.db"
```

指定したパスにスペースがない場合、引用符は必要ありません。

## サーバー停止ユーティリティ (dbstop)

データベースまたはデータベースサーバーを停止します。

### 構文

**dbstop** [ *options* ] [ *server-name* ]

| オプション | 説明 |
|---|---|
| *@data* | 指定された環境変数または設定ファイルからオプションを読み込みます。「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル非表示ユーティリティを使用して、設定ファイルの内容を読みにくくすることができます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |

| オプション | 説明 |
| --- | --- |
| **-c** *"keyword=value; ..."* | 接続文字列を指定します。ネットワークサーバーを停止する場合は、サーバーを停止するパーミッションのあるユーザー ID を接続文字列に含める必要があります。デフォルトでは、ネットワークサーバーに対しては DBA 権限が必要であり、パーソナルサーバーはすべてのユーザーが停止できます。ただし、-gk サーバーオプションを使用するとこれを変更できます。<br><br>接続パラメーターを指定する場合は、サーバー名を指定しないでください。「接続パラメーター」320 ページ、「Unconditional (UNC) 接続パラメーター」368 ページ、「-gk dbeng12/dbsrv12 サーバーオプション」235 ページを参照してください。 |
| **-d** | データベースサーバーは停止しません。代わりに、接続文字列で指定したデータベースだけを停止します。-gd サーバーオプションは、データベースファイルの停止に必要なパーミッションを制御します。「-gd dbeng12/dbsrv12 サーバーオプション」233 ページを参照してください。 |
| **-o** *filename* | 指定したファイルに、出力メッセージを書き込みます。 |
| **-q** | クワイエットモードで実行します (メッセージを表示しません)。 |
| **-x** | サーバーへのアクティブな接続がある場合、サーバーを停止しません。このオプションを使用すると、dbstop はアクティブな接続があるときでもプロンプトを表示しません。 |
| **-y** | サーバーにアクティブな接続がある場合でも、サーバーを停止します。これは、接続パラメーターに Unconditional=YES を含めることと同等です。 |

| オプション | 説明 |
| --- | --- |
| *server-name* | 現在のコンピューター上で稼働中のデータベースサーバーの名前を指定します。シャットダウン時にパーミッションが一切必要にならないように、データベースサーバーを起動してください。パーソナルデータベースサーバーは、デフォルトではこのモードで起動します。ネットワークデータベースサーバーの場合は、**-gk all** オプションを指定してください。「-gk dbeng12/dbsrv12 サーバーオプション」235 ページを参照してください。<br><br>サーバー名を指定する場合は、接続パラメーターを指定しないでください。 |

## 備考

サーバー停止ユーティリティは、データベースサーバーを停止します。-d オプションを使用して、指定したデータベースを停止できます。

サーバー停止ユーティリティはコマンドプロンプトでのみ実行できます。ウィンドウベースの環境では、データベースサーバーメッセージウィンドウで **[シャットダウン]** をクリックしてデータベースサーバーを停止できます。

オプションを使用して、アクティブな接続がある場合にも接続を停止するかどうか、またサーバーを停止するのかデータベースのみを停止するのかを制御できます。

サーバー上にアクティブな接続がある場合の dbstop の動作を制御できます。アクティブな接続がある場合、dbstop はそのサーバーを停止するかどうかをたずねるプロンプトを表示します。この動作を変更するには、-x オプションと -y オプションを使用します。

データベースサーバーを停止できる場合、すべてのデータベースが終了し、データベースサーバーが停止して、同じ名前のサーバーを起動して同じデータベースを実行できる状態になってから、dbstop が完了します。dbstop が完了しても、データベースサーバープロセスは動作したままとなることがあり、そのリソース (-o サーバーオプションで指定された出力ファイルなど) も使用中の状態が続くことがあります。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

dbstop で SQLCONNECT 環境変数を使用する場合は、-c オプションを指定する必要があります。それ以外の場合に -c オプションを使用すると、予期しない結果となることがあります。

## 参照

- 「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』
- 「STOP SERVER 文」『SQL Anywhere サーバー SQL リファレンス』

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### 例

myserver という名前のサーバーを実行していて、データベースは起動していないとします。このサーバーを停止するには、DatabaseName (DBN) 接続パラメーターとしてユーティリティデータベースを指定します。

```
dbstop -c "UID=DBA;PWD=sql;Server=myserver;DBN=utility_db"
```

myserver という名前のサーバーを実行していて、データベース *demo.db* が動作中であるとします。このサーバーとデータベースを停止するには、次のように指定します。

```
dbstop -c "UID=DBA;PWD=sql;Server=myserver"
```

myserver という名前のパーソナルサーバーを実行しているとします。接続が確立されている状態で、このサーバーとデータベースを停止するには、次のように指定します。

```
dbstop -y myserver
```

myserver という名前のサーバーを実行していて、データベース *demo.db* が動作中であるとします。他のデータベースやサーバーを停止しないでデータベース demo.db だけを停止するには、次のコマンドを実行します。

```
dbstop -c "UID=DBA;PWD=sql;Server=myserver;DBN=demo" -d
```

## サポートユーティリティ (dbsupport)

エラーやソフトウェアの使用状況に関する情報を iAnywhere Solutions に送信します。

### 構文

**dbsupport** [ *options* ] *operation* [ *operation-specific-option* ]

**dbsupport** *configuration-options*

| オプション | 説明 |
|---|---|
| *@data* | 指定された環境変数または設定ファイルからオプションを読み込みます。「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル非表示ユーティリティを使用して、設定ファイルの内容を読みにくくすることができます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |
| **-o** *filename* | 指定されたファイルに出力を送信します。 |
| **-q** | 重要なメッセージだけを表示します。 |

| 操作 | 説明 |
| --- | --- |
| **-e** *configuration-option* | 指定された設定オプションの設定を表示します。たとえば次のコマンドを実行して、可能な場合にはプロンプトを表示するように dbsupport を設定したとします。<br><br>dbsupport -cc promptdefy<br><br>この場合、dbsupport -ecc コマンドを実行すると、次の設定が返されます。<br><br>-cc "promptdefy"<br><br>設定ファイルの現在の値が明示的に設定されていない場合は、エコーされません。 |
| **-is** *submission-ID* [ **-rr** *N* ] | iAnywhere Solutions に送信されたクラッシュレポートのステータスをチェックします。<br><br>たとえば、次のコマンドは送信 ID 66 のステータスを返します。<br><br>dbsupport -is 66 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 操作 | 説明 |
| --- | --- |
| **-iu** [ **-r** *N* ] | 使用している SQL Anywhere のビルドに対する更新をチェックします。次の更新があります。<br><br>● **EBF**　Express Bug Fix は、1 つ以上のバグフィックスが含まれる、ソフトウェアのサブセットです。これらのバグフィックスは、更新のリリースノートにリストされます。バグフィックス更新を適用できるのは、同じバージョン番号を持つインストール済みのソフトウェアに対してだけです。ソフトウェアでテストを行っている場合は、そのソフトウェアでアプリケーションを徹底的にテストしないかぎり、アプリケーションとともにこれらのファイルを配布しないでください。<br><br>● **メンテナンスリリース**　メンテナンスリリースは、同じメジャーバージョン番号を持つ旧バージョンのインストール済みソフトウェアをアップグレードするための完全なソフトウェアセットです (バージョン番号のフォーマットは、*メジャー.マイナー.パッチ.ビルド*です)。バグフィックスとその他の変更については、アップグレードのリリースノートにリストされます。<br><br>更新については、Interactive SQL と Sybase Central を使用して確認することもできます。「ソフトウェアの更新」907 ページを参照してください。 |
| **-lc** | iAnywhere Solutions に送信されていないすべてのクラッシュレポートのリストを作成します。リストされるレポート名には、-sc オプションを使用できます。 |

| 操作 | 説明 |
| --- | --- |
| **-ls** | iAnywhere Solutions に送信されたすべてのレポートの送信 ID のリストを作成します。次に例を示します。<br><br>dbsupport -ls<br><br>このコマンドは次のような情報を返します。<br><br>SQL Anywhere Support Utility Version 12.0.1.2040<br>Submission ID: 217756 reported: Wed Jul 30 15:54:43 2011<br>Submission ID: 217757 reported: Wed Jul 30 15:57:07 2011<br>Submission ID: 217759 reported: Wed Jul 30 16:14:24 2011<br>Submission ID: 217760 reported: Wed Jul 30 16:14:30 2011<br>Submission ID: 217761 reported: Wed Jul 30 16:28:12 2011<br>Submission ID: 221030 reported: Mon Sep 08 12:34:59 2011<br>Submission ID: 221031 reported: Mon Sep 08 12:35:05 2011 |
| **-pc** *filename* | クラッシュレポートの情報を表示します。このオプションを使用すると、情報を確認してからレポートを iAnywhere Solutions に送信できます。 |
| **-pd** | 収集された診断情報を表示します。このオプションを使用すると、情報を確認してからレポートを iAnywhere Solutions に送信できます。 |
| **-ps** *submission-ID* | iAnywhere Solutions に送信された特定のレポートに関する情報を表示します。次に例を示します。<br><br>dbsupport -ps 4<br><br>このコマンドは送信 ID 4 の情報を返します。<br><br>SQL Anywhere Support Utility Version 12.0.1.2040<br>Submission ID: 217756 reported: Wed Jul 30 15:54:43 2011 |
| **-sa** [ **-r** *number-of-submission-retries* ] | 診断ディレクトリに保存されているすべてのクラッシュレポートと診断情報を、iAnywhere Solutions に送信します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 操作 | 説明 |
| --- | --- |
| **-sc** *reportname* [ **-r** *number-of-submission-retries* ] [ **-nr** \| **-rr** *N* ] | クラッシュレポートと診断情報を iAnywhere Solutions に送信します。次に例を示します。<br>dbsupport -sc SA12_20051220_133828_32116<br>-lc オプションを使用すると、送信されていないレポートのリストを参照できます。 |
| **-sd** [ **-r** *number-of-submission-retries* ] | 診断情報だけを iAnywhere Solutions に送信します。<br>診断ディレクトリの詳細については、「SADIAGDIR 環境変数」449 ページを参照してください。 |

| 設定オプション | 説明 |
| --- | --- |
| **-cc** [ **autosubmit** \| **no** \| **promptDefY** \| **promptDefN** ] | dbsupport によるプロンプトの動作を変更します。次のいずれかのオプションを指定できます。<br><br>● **autosubmit**　レポートを自動送信します。<br><br>● **no**　レポートを送信してよいかどうかを確認しません。レポートと機能の統計は送信されません。<br><br>● **promptDefY**　可能な場合、レポートを送信してよいかどうかを確認します。回答がない場合は、レポートを送信します。<br><br>● **promptDefN**　可能な場合、レポートを送信してよいかどうかを確認します。回答がない場合は、レポートを送信しません。これはデフォルトの動作です。<br><br>たとえば、アプリケーションで Embedded SQL Anywhere を使用する場合は、iAnywhere Solutions にレポートを送信しないようにサポートユーティリティを設定することもできます。<br><br>このオプションを指定すると、その値がデフォルトとしてサポートユーティリティに使用されます。設定は、診断ディレクトリ内の *dbsupport.ini* ファイルに保存されます。<br><br>次のコマンドは、レポートを送信せず、ユーザーに対してレポート送信の確認メッセージを表示しないように、サポートユーティリティを設定します。<br><br>dbsupport -cc no |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 設定オプション | 説明 |
| --- | --- |
| **-cd** *retry-delay* | レポートの送信が失敗した場合にリトライするまでの遅延時間を秒単位で指定します。デフォルトの遅延時間は 30 秒です。<br><br>このオプションを指定すると、その値がデフォルトとしてサポートユーティリティに使用されます。設定は、診断ディレクトリ内の *dbsupport.ini* ファイルに保存されます。<br><br>次の dbsupport コマンドでは、送信が失敗した場合に 3 秒間隔で最大 4 回リトライするように指定しています。<br><br>dbsupport -cr 4 -cd 3 |
| **-ce** *email-address*;*email-server*[:*port* ][;*user-id*:*password* ] | クラッシュの発生後に電子メールを送信するアドレスを指定します。電子メールは、*email-server* SMTP サーバーを使用して送信されます。必要に応じて、使用するポートや、SMTP サーバーでの認証に使用する *user-id* と *password* を指定できます。 |
| **-cet** | -ce オプションで指定した電子メール設定をテストします。 |

| 設定オプション | 説明 |
| --- | --- |
| **-ch** *crash-handler-program* | クラッシュが発生したときに呼び出すプログラムを指定します。<br><br>このオプションを指定すると、その値がデフォルトとしてサポートユーティリティに使用されます。設定は、診断ディレクトリ内の *dbsupport.ini* ファイルに保存されます。<br><br>データベースサーバーは、*crash-handler-program* に渡す情報を構築する 3 つの代入パラメーターをサポートしています。<br><br>● **%F**　このパラメーターは、生成されたレポートファイルのフルパスに置き換えられます。<br><br>● **%P**　このパラメーターは、レポートを生成したプログラムの名前に置き換えられます。たとえば、バージョン 12 のパーソナルデータベースサーバーがレポートを生成した場合は、dbeng12 が返されます。<br><br>● **%S**　このパラメーターは、クラッシュまたは致命的なエラーが発生したときに使用されていたデータベースサーバーの名前に置き換えられます。たとえば、Sample というデータベースサーバーがレポートを生成した場合は、Sample が返されます。<br><br>%F、%P、%S の代わりに $F、$P、$S を使用できます。% と $ の解釈がコマンドシェルによって異なるため、両方の文字を提供しています。たとえば、4NT では %F が環境変数 F の値に置き換えられるため、それを回避するために $F を使用できます。<br><br>*c:¥test.bat* にクラッシュハンドラープログラムがあり、次のコマンドが含まれているとします。<br><br>copy %1 c:¥archives<br>echo %2<br><br>Windows では、クラッシュが発生した場合、次のコマンドによって dbsupport が *c:¥test.bat* を起動し、2 つのパラメーターを渡します。レポートを送信する場合は、このプログラムが呼び出された後でレポートが送信されます。<br><br>dbsupport -ch "c:¥test.bat ¥"%F¥" parm2" |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 設定オプション | 説明 |
| --- | --- |
| | %F に指定されたパスが 1 つ目のパラメーターとして *c:¥test.bat* に渡されます。2 つ目のパラメーターとして、*parm2* が *c:¥test.bat* に渡されます。引数を取るクラッシュハンドラープログラムを指定する場合は引用符が必要です。<br><br>上記の例では、生成されたレポートファイルのフルパスを引用符で囲んでいます。dbsupport が使用中であるためにレポートファイルにアクセスできなくなるという問題を回避するために、クラッシュハンドラープログラムはレポートファイルの専用コピーを作成する必要があります。 |
| **-ch-** | *dbsupport.ini* ファイルに格納されているクラッシュハンドラー設定を削除します。次に例を示します。<br><br>dbsupport -ch- |
| **-cid** *customer-id* | 送信レポート内でユーザー自身を識別するための文字列を指定します。このオプションを指定すると、その値がデフォルトとしてサポートユーティリティに使用されます。設定は、診断ディレクトリ内の *dbsupport.ini* ファイルに保存されます。<br><br>次の例は、dbsupport に対してカスタマー ID 文字列を指定しています。<br><br>dbsupport -cid myid@company.com<br><br>dbsupport -cid "MyClientApp 1.0" |
| **-cid-** | *dbsupport.ini* ファイルからカスタマー ID 文字列を削除します。次に例を示します。<br><br>dbsupport -cid- |

| 設定オプション | 説明 |
| --- | --- |
| **-cp** { *email-server* [ **:***port* ] \| **autodetect** } | エラーレポートの送信に使用する HTTP プロキシのホストとポートを設定します。<br><br>Windows では、-cp autodetect の構文もサポートされます。このオプションを指定し、Internet Explorer でプロキシサーバーとポートが設定され、現在有効である場合、dbsupport はそのシステム設定に基づいてプロキシサーバーとポートを設定します。Internet Explorer で、プロキシサーバーとポートを設定しなければならない場合があります。 |
| **-cp-** | *dbsupport.ini* ファイルから HTTP プロキシのホストとポートの設定を削除します。次に例を示します。<br><br>dbsupport -cp- |
| **-cr** *number-of-submission-retries* | 送信が失敗した場合のリトライ回数を指定します。<br><br>このオプションを指定すると、その値がデフォルトとしてサポートユーティリティに使用されます。設定は、診断ディレクトリ内の *dbsupport.ini* ファイルに保存されます。<br><br>次の dbsupport コマンドでは、操作が失敗した場合に 3 秒間隔で最大 4 回リトライするように指定しています。<br><br>dbsupport -cr 4 -cd 3 |

| 操作固有のオプション | 説明 |
| --- | --- |
| **-ac** | 送信にコメントを追加します。 |
| **-af** | 送信にファイルを付加します。-af オプションを複数回指定すると、送信に複数のファイルを付加できます。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 操作固有のオプション | 説明 |
| --- | --- |
| **-nr** | dbsupport がサーバーで送信のステータスをチェックしないことを指定します。たとえば、次のコマンドは、レポートを送信するだけで、その送信のステータスについてはチェックしません。<br>dbsupport -nr -sc SA12_20051220_133828_32116<br>デフォルトでは、送信しようとしている問題について解決策がすでに提示されていないかどうかチェックします。 |
| **-r** *number-of-submission-retries* | dbsupport がレポートの送信をリトライする最大回数を指定します。0 を指定すると、リトライ回数は無限となります。デフォルト値は 10 です。*dbsupport.ini* ファイルに -cr オプションが保存されている場合、-r を指定すると -cr の値は上書きされます。 |
| **-rd** *retry-delay* | レポートを再送するまでの dbsupport の待ち時間を秒単位で指定します。デフォルト値は 30 です。*dbsupport.ini* ファイルに -cd オプションが保存されている場合、-rd を指定すると -cr の値は上書きされます。 |
| **-rr** *number-of-submission-response-retries* | dbsupport が送信に対する応答の取得をリトライする最大回数を指定します。0 を指定すると、リトライ回数は無限となります。デフォルト値は 10 です。 |

**備考**

SQL Anywhere プログラムで致命的なエラーまたはクラッシュが発生した場合は、エラー発生時のプログラムに関する技術情報を含むエラーレポートが作成されます。エラーレポートが作成されると、このレポートを SQL Anywhere サポートチームに提出するよう要求されます。サポートユーティリティ (dbsupport) を使用して、このレポートを提出することもできます。エラーレポートを送信すると、致命的なエラーやアサーションの原因の究明に役立つことがあります。エラーレポートが提出されると、テクニカルサポートの問い合わせで使用できる提出 ID を受け取ります。問題をテクニカルサポート部門に報告すると、このエラーファイルを送信することで、テクニカルサポート部門が問題をすばやく処理することができます。テクニカルサポートに連絡しない場合でも、製品の改良に役立てるために、エラーレポートを送信することをおすすめします。

サポートユーティリティ (dbsupport) では、次のタスクを実行できます。

●診断情報とクラッシュレポートをインターネット経由で iAnywhere Solutions に送信する

●機能の統計を送信する

●送信済みおよび未送信のクラッシュレポートに関する情報を表示する

●送信済みおよび未送信のクラッシュレポートに関する情報を出力する

●送信のステータスを問い合わせる

●使用中の SQL Anywhere のビルドに対応するソフトウェア更新のリリース状況を問い合わせる

●致命的なエラー (アサーション／クラッシュ) を検出した場合の dbsupport の動作を設定する。たとえば、データベースサーバーがエラーレポートを送信するたびに、指定したハンドラープログラムを実行するように設定できます。この機能は、エラー処理プロセスにカスタムアクションを追加する場合に便利です。

●レポートの再送回数を設定する。たとえば、レポートの送信については、リトライの間隔を 30 秒、最大回数を 10 回に設定することもできます。この機能によって、サービスが一時的に使用不能となった場合の対処方法を組み込むことができます。

サポートユーティリティの設定は、診断ディレクトリの *dbsupport.ini* ファイルに格納されます。

操作固有のオプションは、*dbsupport.ini file* に保存されている動作を含め、デフォルトの動作に優先させる場合に役立ちます。

グラフィカルな管理ツールで、エラーレポートを送信するオプションを削除するよう設定することができます。 「管理ツールの設定」『SQL Anywhere サーバー プログラミング』を参照してください。

エラーレポートを送信しないように選択すると、レポートの内容はハードディスクの診断ディレクトリに残ります。このディレクトリは SQL Anywhere プログラムによって自動的に作成され、エラーレポートの保存と、サポートユーティリティ (dbsupport) の設定の格納にのみ使用されます。

**診断ディレクトリの場所**　診断ディレクトリの場所は、プラットフォームによって異なります。

○ **Windows (Windows Mobile を除く)**

1. SADIAGDIR 環境変数で指定されたディレクトリ (設定している場合)。
2. デフォルトの診断ディレクトリ：

   ●Windows XP：*%ALLUSERSPROFILE%¥Application Data¥SQL Anywhere 12¥diagnostics*

   ●Windows Vista 以降：*%ALLUSERSPROFILE%¥SQL Anywhere 12¥diagnostics*
3. 現在のディレクトリ。
4. テンポラリディレクトリ。SATMP 環境変数と、TMP、TMPDIR、TEMP 環境変数を参照してください。

Windows では、診断ディレクトリがコンピューターのすべてのユーザーが書き込み可能なロケーションにあります。コンピューターに複数のユーザーアカウントがある場合で、あるユーザーのエラーレポートを他のユーザーから分離する場合は、各ユーザーに SADIAGDIR

環境変数を設定して、読み取りが可能なディレクトリを指定し、書き込みは特定のユーザーのみに限定します。 「SADIAGDIR 環境変数」449 ページを参照してください。

○ **Windows Mobile**

1. 実行可能ファイルのディレクトリ、またはそのディレクトリが書き込み可能でない場合。
2. 現在のディレクトリ、またはそのディレクトリが書き込み可能でない場合。
3. テンポラリディレクトリ。Windows Mobile でのレジストリ設定を参照してください。

○ **UNIX、Linux、OS X**

1. SADIAGDIR 環境変数で指定されたディレクトリ (設定している場合)。
2. デフォルトの診断ディレクトリ：*$HOME/.sqlanywhere12/diagnostics*。
3. 現在のディレクトリ。
4. テンポラリディレクトリ。SATMP 環境変数と、TMP、TMPDIR、TEMP 環境変数を参照してください。

**注意**
UNIX では、データベースサーバーまたは Mobile Link サーバーをデーモンとして実行する場合やユーザーが root/nobody である場合、ユーザーのホームディレクトリにクラッシュレポートを書き込むことは推奨されません。

**エラーレポートファイル名**　エラーレポートのファイル名は、次のように構成されます。

○アプリケーションを識別するプレフィクス

| アプリケーションプレフィクス | アプリケーション |
|---|---|
| DBCONSOLE | SQL Anywhere コンソールユーティリティ |
| ISQL | Interactive SQL ユーティリティ |
| LSN | Listener ユーティリティ |
| MLC | Mobile Link クライアント |
| MLS | Mobile Link サーバー |
| ML_MON | Mobile Link モニター |
| QAA | QAnywhere Agent |
| SA | パーソナルデータベースサーバーまたは<br>ネットワークデータベースサーバー |

| アプリケーションプレフィクス | アプリケーション |
|---|---|
| SCJ | Sybase Central |
| SR | SQL Remote |

○ソフトウェアのバージョンを示す数値

○アンダースコアで連結された 2 つのフィールド (エラーレポートが作成されたときのタイムスタンプを示す)

○アプリケーションの識別子

○拡張子 *.dmp*

たとえば、*SA12_20110620_133828_32116.dmp* は、SQL Anywhere バージョン 12 のデータベースサーバーで、2011/06/20 の 1:38:28 に、プロセス 32116 から作成されたエラーレポートであることを示します。

**参照**

● 「SADIAGDIR 環境変数」449 ページ

**例**

この dbsupport のデフォルト動作を変更するには、-cc オプションを使用します。

●次のコマンドは、ユーザーにメッセージを表示せず、自動的にエラーレポートを送信するように dbsupport を設定します。

```
dbsupport -cc autosubmit
```

●次のコマンドは、自動的なエラーレポート送信を無効にします。

```
dbsupport -cc no
```

-lc オプションを使用してエラーレポートのリストを表示できます。

●次のコマンドは、iAnywhere Solutions に送信されていないすべてのクラッシュレポートのリストを作成します。

```
dbsupport -lc
```

-sa、-sc、または -sd オプションを使用してエラーレポートを手動で送信できます。

●次のコマンドは、診断ディレクトリに保存されているすべてのクラッシュレポートと診断情報を、iAnywhere Solutions に送信します。

```
dbsupport -sa
```

-ac オプションと -af オプションをそれぞれ指定して、エラーレポートと共にコメントとファイルを送信することもできます。

```
dbsupport -sc SA12_20080901_113308_3360 -ac "The message.txt file provides more
information about this error report." -af c:¥scenario.txt -af c:¥message.txt
```

# トランザクションログユーティリティ (dblog)

データベースのトランザクションログを管理します。

## 構文

**dblog** [ *options* ] *database-file*

| オプション | 説明 |
|---|---|
| **@***data* | 指定された環境変数または設定ファイルからオプションを読み込みます。「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル非表示ユーティリティを使用して、設定ファイルの内容を読みにくくすることができます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |
| **-ek** *key* | 強力に暗号化されているデータベースの暗号化キーをコマンドに直接指定します。データベースが強力に暗号化されている場合、データベースまたはトランザクションログを使用するには必ず暗号化キーを指定します。高度に暗号化されたデータベースの場合、-ek または -ep のどちらかを指定します。両方同時には指定できません。強力に暗号化されたデータベースでは、正しいキーを指定しないとコマンドが失敗します。 |
| **-ep** | 暗号化キーの入力を求めるプロンプトを表示するよう指定します。このオプションを指定すると、暗号化キーを入力するためのウィンドウが表示されます。クリアテキストでは暗号化キーを見ることができないようにすることで、高いセキュリティが得られます。高度に暗号化されたデータベースの場合、-ek または -ep のどちらかを指定します。両方同時には指定できません。強力に暗号化されたデータベースでは、正しいキーを指定しないとコマンドが失敗します。 |

| オプション | 説明 |
| --- | --- |
| **-ir** | このオプションは、delete_old_logs オプションのために保存されている SQL Remote ログオフセットをリセットし、必要のないトランザクションログを削除できるようにします。このオプションは、このデータベースで SQL Remote を使用しなくなったが、Mobile Link 同期は引き続き使用する場合に指定します。 |
| **-is** | このオプションは、delete_old_logs オプションのために保存されている Mobile Link ログオフセットをリセットし、必要のないトランザクションログを削除できるようにします。このオプションは、このデータベースで Mobile Link 同期を使用しなくなったが、SQL Remote は引き続き使用する場合に指定します。 |
| **-m** *mirror-name* | 新しいトランザクションログミラーのファイル名を設定します。データベースがトランザクションログミラーを現在使っていない場合、データベースはこの設定された名前を使って起動します。すでにトランザクションログミラーを使っている場合、データベースはトランザクションログミラーとして新しいファイル名を使うように変更します。 |
| **-n** | トランザクションログとトランザクションログミラーの使用を停止します。トランザクションログを使用しないと、データベースはデータレプリケーションに参加できず、またはデータリカバリにトランザクションログを使えません。SQL Remote または dbmlsync トランケーションオフセットが存在する場合は、対応する無視オプション (SQL Remote には -ir、dbmlsync には -is) も指定されていないかぎり、トランザクションログを削除することはできません。データベースで監査が有効になっている場合、トランザクションログの使用を停止することはできません (まず監査を無効にする必要があります)。 |
| **-o** *filename* | 指定したファイルに、出力メッセージを書き込みます。 |
| **-q** | クワイエットモードで実行します (メッセージを表示しません)。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| **-r** | トランザクションログミラーを管理しているデータベースで、1 つのトランザクションログを管理します。 |
| **-t** *log-name* | 新しいトランザクションログのファイル名を設定します。データベースがトランザクションログを現在使っていない場合、データベースは設定されたファイル名を使って起動します。すでにトランザクションログを使っている場合、データベースはトランザクションログとして新しいファイル名を使うように変更します。 |
| **-x** *n* | トランザクションログの現在の相対オフセットを *n* にリセットし、データベースがレプリケーションに参加できるようにします。このオプションは、SQL Remote 統合データベースを再ロードするときに使用します。「再ロードファイルへのリモートデータベースの抽出」『SQL Remote』を参照してください。 |
| **-z** *n* | トランザクションログの現在の開始オフセットを *n* にリセットし、データベースがレプリケーションに参加できるようにします。このオプションは、SQL Remote 統合データベースを再ロードするときに使用します。「再ロードファイルへのリモートデータベースの抽出」『SQL Remote』を参照してください。 |

**備考**

dblog ユーティリティで、データベースに関連するトランザクションログまたはトランザクションログミラーの名前を表示、変更できます。データベースがトランザクションログやミラーを管理するのを停止したり、開始したりできます。

トランザクションログミラーはトランザクションログの重複コピーであり、データベースによって並列に管理されています。

データベースを初期化するときに、トランザクションログの名前を最初に設定します。トランザクションログユーティリティは、データベースファイルを処理します。トランザクションログファイル名を変更するときは、データベース上でデータベースサーバーを実行しないでください (実行するとエラーメッセージが表示されます)。

このユーティリティは、トランザクションログに関する次のような追加情報も表示します。

- バージョン番号
- トランザクションログファイルの名前
- トランザクションログミラーファイルの名前 (トランザクションログミラーがある場合)
- 現在の相対オフセット

次の方法で、トランザクションログユーティリティにアクセスできます。

●Sybase Central の**ログファイル設定の変更ウィザード**を使用する。

●Interactive SQL から ALTER DATABASE *dbfile* ALTER LOG 文を使用する。

●コマンドプロンプトで、dblog コマンドを入力する。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

**参照**

- 「トランザクションログの場所の変更」129 ページ
- 「ALTER DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』

# アンロードユーティリティ (dbunload)

SQL スクリプトファイルにデータベースをアンロードします。

**構文**

**dbunload** [ *options* ] [ *directory* ]

| オプション | 説明 |
|---|---|
| *@data* | 指定された環境変数または設定ファイルからオプションを読み込みます。「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル非表示ユーティリティを使用して、設定ファイルの内容を読みにくくすることができます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |

| オプション | 説明 |
| --- | --- |
| **-ac** *"keyword=value; ..."* | 既存のデータベースに接続し、データをそのデータベースに直接再ロードすることによって、データベースのアンロードとその結果を既存のデータベースに再ロードする操作を一度に実行します。このオプションは、Windows Mobile ではサポートされていません。<br><br>たとえば、dbinit ユーティリティを使用して新しいデータベースを作成し、このオプションを使用して再ロードすることもできます。この方法は、初期化オプションを変更する場合に便利です。<br><br>次のコマンド (すべて 1 行に入力) は、*c:¥mydata.db* データベースのコピーを *c:¥mynewdata.db* という既存のデータベースファイルにロードします。<br><br>dbunload -c "UID=DBA;PWD=sql;DBF=c:¥mydata.db"<br>-ac "UID=DBA;PWD=sql;DBF=c:¥mynewdata.db"<br><br>新しいデータベースが元のデータベースと同じファイル名を持つ必要がある場合、DBN 接続パラメーターを使用して正しいデータベース名を指定します。<br>「DatabaseName (DBN) 接続パラメーター」336 ページを参照してください。<br><br>次のコマンド (すべてを 1 行で入力する) は、同じファイル名 *mydata.db* を持つ新しいデータベースに古いデータベースのコピーをロードします。<br><br>dbunload -c "UID=DBA;PWD=sql;DBN=old;DBF=c:¥old_directory¥mydata.db"<br>-ac "UID=DBA;PWD=sql;DBN=new;DBF=c:¥new_directory¥mydata.db"<br><br>元のデータベースがバージョン 9 以前の SQL Anywhere で作成されており、新しいデータベースがまだ実行されていない場合は、-ac オプションでデータベースサーバーの名前を指定する必要があります。次に例を示します。<br><br>dbunload -c "UID=DBA;PWD=sql;DBF=c:¥mydata.db"<br>-ac "UID=DBA;PWD=sql;DBF=c:¥mynewdata.db;Server=newserver"<br><br>このオプションを使用した場合、データの中間コピーはディスク上に作成されないため、コマンドにアンロードディレクトリは指定しないでください。このため、データのセキュリティが向上します。 |

| オプション | 説明 |
| --- | --- |
| **-an** *database* | このオプションを使用すると、データベースのアンロード、新規データベースの作成、データのロードを一度に実行できます。このオプションは、Windows Mobile 上、または Intel ベースの Mac OS X でバージョン 9 以前のデータベースを再構築する場合にはサポートされません。<br><br>ソースデータベースを作成したときに指定されたオプションが、新しいデータベースの作成に使用されます。ただし、サポートされているその他の dbunload オプション (たとえば、ページサイズを変更するときは -ap、テーブル暗号化を有効にするときは -et) を指定して、必要に応じて初期化オプションを変更することができます。<br><br>たとえば、次のコマンド (すべて 1 行に入力) は、*mydatacopy.db* という名前の新規データベースファイルを作成し、*mydata.db* のスキーマとデータをその中にコピーします。<br><br>dbunload -c "UID=DBA;PWD=sql;DBF=c:¥mydata.db"<br>-an c:¥mydatacopy.db<br><br>このオプションを使用した場合、データの中間コピーはディスク上に作成されないため、コマンドにアンロードディレクトリは指定しません。このため、データのセキュリティが向上します。<br><br>新しいデータベースが作成されると、DB 領域ファイル名に R が追加され、元のデータベースの DB 領域と同じディレクトリに新しいデータベースの DB 領域が作成された場合の名前の競合を防ぎます。たとえば、アンロードされたデータベースの *library.db* ファイルに library という DB 領域がある場合、新しいデータベースの library DB 領域は *library.dbR* となります。<br><br>-an には、ファイル名としてデータベースサーバーに対する相対パスを指定します。 |
| **-ap** *size* [ **k** ] | 新しいデータベースのページサイズを設定します。-an または -ar も一緒に指定しないと、このオプションは無視されます。データベースのページサイズには、2048、4096、8192、16384、32768 バイトのいずれかを指定できます。デフォルトは元のデータベースのページサイズです。**k** を使用して、キロバイトの単位を指定します (-ap 4k など)。このオプションとともに -an または -ar のいずれかを指定する必要があります。データベースサーバーですでにデータベースが実行中の場合、サーバーのページサイズ (-gp オプションで設定) は新規ページサイズを処理するのに十分な容量でなければなりません。「-gp dbeng12/dbsrv12 サーバーオプション」242 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| **-ar** [ *directory* ] | 古いデータベースと同じ設定を持つ新しいデータベースを作成し、新しいデータベースを再ロードして、古いデータベースと置き換えます。ただし、サポートされているその他の dbunload オプション (たとえば、ページサイズを変更するときは -ap、テーブル暗号化を有効にするときは -et) を指定して、必要に応じて初期化オプションを変更することができます。<br><br>このオプションを使用する場合は、データベースへの他の接続がなく、データベース接続が、ネットワークではなくローカルである必要があります。このオプションは、Windows Mobile 上、または Intel ベースの Mac OS X でバージョン 9 以前のデータベースを再構築する場合にはサポートされません。<br><br>オプションの *directory* を指定すると、レプリケーションを行うためにトランザクションログのオフセットがリセットされ、古いデータベースのトランザクションログが指定のディレクトリに移動されます。指定されたディレクトリは、Message Agent が使用する古いトランザクションログを保持している必要があります。トランザクションログ管理は、データベースがレプリケーションで使用されるときだけ実行されます。SQL Remote パブリッシャーがない場合、古いトランザクションログは不要なので、指定するディレクトリにはコピーされず、削除されます。「同期やレプリケーションに関連するデータベースのバックアップ」1107 ページを参照してください。<br><br>新しいデータベースが作成されると、DB 領域ファイル名に R が追加され、元のデータベースの DB 領域と同じディレクトリに新しいデータベースの DB 領域が作成された場合の名前の競合を防ぎます。たとえば、アンロードされたデータベースの *library.db* ファイルに library という DB 領域がある場合、新しいデータベースの library DB 領域は *library.dbR* となります。<br><br>暗号化されたデータベースを再構築する場合は、新しいデータベースで元のデータベースと同じ暗号化キーを使用する必要があります。<br><br>-ar オプションを使用すると、データベースのトランケーションポイントが 0 にリセットされます。 |
| **-c** *"keyword=value; ..."* | ソースデータベースの接続パラメーターを指定します。「接続パラメーター」320 ページを参照してください。<br><br>ユーザーはデータベースの全テーブル上にパーミッションを持っている必要があるため、user ID には DBA 権限を持つユーザー ID を指定してください。<br><br>たとえば、次の文は、パスワード sql を指定してユーザー ID DBA として接続し、サンプルデータベースをアンロードします。データは *c:¥unload* ディレクトリにアンロードされます。<br><br>dbunload -c "DBF=%SQLANYSAMP12%¥demo.db;UID=DBA;PWD=sql" c:¥unload |

| オプション | 説明 |
|---|---|
| **-cm** { **sql** \| **dbinit** } | サーバーメッセージウィンドウに、アンロードしているデータベースと同じデータベースを作成する CREATE DATABASE 文または dbinit コマンドを表示します。-an も一緒に指定した場合、表示されるコマンドは、新しいデータベースを作成するコマンドです。<br><br>● **sql** *reload.sql* ファイルに書き込まれる CREATE DATABASE 文を表示します。<br><br>● **dbinit** dbinit ユーティリティコマンドを表示します。<br><br>既存の強力に暗号化された (-an を指定していない) データベースの文またはコマンドを表示するとき、データベースから暗号化キーを取得できないので、ENCRYPTED 句または -ek オプションに疑問符 (?) が表示されます。<br><br>バージョン 10 以前のデータベースサーバーで作成したデータベースをアンロードする場合は、データベースを作成するコマンドまたは文は表示されません。 |
| **-cp** | 実行される UNLOAD TABLE 文に COMPRESSED キーワードを追加して、テーブルデータ出力ファイルを圧縮します。このオプションは、-an または -ar が指定されている場合は機能しません。 |
| **-d** | データベース定義文 (CREATE TABLE、CREATE INDEX など) は生成されません。*reload.sql* にはデータを再ロードするための文のみが記述されます。 |
| **-dc** | データベース内のすべての計算カラムを再計算します。デフォルトでは、計算カラムは再計算されません。-dc オプションを指定すると、*reload.sql* スクリプトに計算カラムの再計算に対応するセクションが追加されます。追加される文は次のような形式です。<br><br>ALTER TABLE "owner"."table-name"<br>ALTER "computed-column" SET COMPUTE (compute-expression);<br><br>テーブルに CURRENT DATE などのコンテキスト依存の計算カラムがある場合は、-dc オプションを使用するのではなく、ALTER TABLE 文を使用して計算カラムの値を再計算することをおすすめします。「ALTER TABLE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
|---|---|
| **-e** *table, ...* | 指定されたテーブルを *reload.sql* ファイルから除外します。大文字と小文字を区別するデータベースであっても、テーブル名では常に大文字と小文字が区別されません。<br><br>-e オプションを使用して作成した *reload.sql* ファイルにはすべてのデータベーステーブルが含まれるわけではないので、このファイルを使用してデータベースを再構築しないでください。外部キーによって参照されているテーブルがある場合、そのテーブルの内容がなければ、データベースを再構築できません。<br><br>-e オプションを使用する場合は、必ず -d オプションも使用し、-e で指定したテーブルを除くすべてのテーブルのデータをアンロードすることをおすすめします。 |
| **-ea** *algorithm* | データベース暗号化またはテーブル暗号化 (-et) で使用する暗号化アルゴリズムを指定します。単純暗号化の場合は、(-ek や -ep を指定するのではなく) -ea simple を指定します。単純暗号化はデータベースの難読化に相当し、データベースファイルが不用意に直接アクセスされた場合にデータが表示されないようにすることだけを目的としています。これにより第三者は、ディスクユーティリティを使用してファイルを表示し、データベースのデータを解読することが困難になります。<br><br>セキュリティを強化するには、128 ビットの場合は AES、256 ビットの場合は AES256 の強力な暗号化を指定します。FIPS 認定の暗号化を使用するには、128 ビットの場合は AES_FIPS、256 ビットの場合は AES256_FIPS をそれぞれ指定してください。強力な暗号化を使用するためには、-ek または -ep オプションも指定する必要があります。強力な暗号化の詳細については、「強力な暗号化」1397 ページを参照してください。<br><br>暗号化されていないデータベースを作成するには、-ea none を指定するか、-ea オプションを指定しません (-e、-et、-ep、-ek オプションも指定しません)。<br><br>-ea オプションを指定しない場合、デフォルトの動作は次のようになります。<br><br>● -ea none (-ek、-ep、-et が指定されない場合)<br>● -ea AES (-ek または -ep が指定される場合) (-et の指定とは無関係)<br>● -ea simple (-ek または -ep を指定せずに -et を指定する場合)<br><br>アルゴリズム名の大文字と小文字は区別されません。<br><br>**注意**<br>別途ライセンスが必要な必須コンポーネント<br><br>ECC 暗号化と FIPS 認定の暗号化には、別途ライセンスが必要です。強力な暗号化テクノロジはすべて、輸出規制対象品目です。<br><br>「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。 |

| オプション | 説明 |
|---|---|
| **-ek** *key* | (-an オプションを使用して) データベースのアンロードと再ロードを実行する場合に、作成する新しいデータベースに対する暗号化キーを dbunload コマンドに指定します。強力に暗号化されたデータベースを作成する場合、データベースやトランザクションログを使用するには必ず暗号化キーを指定します。データベースの暗号化に使用されるアルゴリズムは、-ea オプションで指定したアルゴリズムです。-ek オプションを指定して、-ea オプションを指定しないと、AES アルゴリズムが使用されます。「強力な暗号化」1397 ページを参照してください。<br><br>キーは保護してください。キーのコピーは安全な場所に保管してください。キーを紛失すると、データベースにまったくアクセスできなくなり、リカバリも不可能になります。 |
| **-ep** | (-an オプションを使用して) データベースのアンロードと再ロードを実行する場合に、作成する新しいデータベースに対する暗号化キーの指定を求めるプロンプトを表示します。クリアテキストでは暗号化キーを見ることができないようにすることで、高いセキュリティが得られます。-an を指定せずに -ep を指定した場合、-ep オプションは無視されます。-ep と -an を指定する場合は、暗号化キーが正確に入力されたことを確認するために、2 回入力してください。キーが一致しない場合は、アンロードは失敗します。「強力な暗号化」1397 ページを参照してください。 |
| **-er** | アンロード中に暗号化されたテーブルの暗号化を解除します。<br><br>テーブル暗号化が有効であるデータベースを再構築するとき、-er または -et を指定して、新しいデータベースのテーブルで暗号化を有効にするかどうかを指定する必要があります。このオプションを指定しないと、データを新しいデータベースにロードしようとしたときにエラーが発生します。<br><br>次のコマンドは、テーブルの暗号化を解除してデータベース (*mydata.db*) をアンロードし、テーブル暗号化が有効になっていない新しいデータベース (*mydatacopy.db*) に再ロードします。<br><br>`dbunload -an c:¥mydatacopy.db -er`<br>`  -c "UID=DBA;PWD=sql;`<br>`  DBF=c:¥mydata.db;`<br>`  DBKEY=29bN8cj1z"` |

| オプション | 説明 |
|---|---|
| **-et** | 新しいデータベースのテーブル暗号化を有効にします (-an または -ar も一緒に指定する必要があります)。-ea オプションを指定せずに -et オプションを指定すると、AES アルゴリズムが使用されます。-et オプションを指定する場合、-ep または -ek も指定しないと操作は失敗します。新しいデータベースのテーブル暗号化設定を変更して、アンロードしているデータベースのテーブル暗号化設定とは異なるものに設定できます。<br><br>テーブル暗号化が有効であるデータベースを再構築するとき、-er または -et を指定して、新しいデータベースのテーブルで暗号化を有効にするかどうかを指定する必要があります。このオプションを指定しないと、データを新しいデータベースにロードしようとしたときにエラーが発生します。<br><br>次の例は、テーブルが単純暗号化アルゴリズムによって暗号化されているデータベース (*mydata.db*) を、テーブル暗号化が有効になっている新しいデータベース (*mydatacopy.db*) にアンロードし、キーを 34jh として AES_FIPS アルゴリズムを使用します。<br><br>dbunload -an c:¥mydatacopy.db -et -ea AES_FIPS<br>-ek 34jh<br>-c "UID=DBA;PWD=sql;DBF=c:¥mydata.db" |
| **-g** | **マテリアライズドビュー**　MANUAL REFRESH と定義されているマテリアライズドビューは、デフォルトでは再ロード後に初期化されません。このようなマテリアライズドビューを再ロードプロセス中に初期化するには、-g オプションを指定してください。-g を指定すると、データベースサーバーによって sa_refresh_materialized_views システムプロシージャーが呼び出されます。「sa_refresh_materialized_views システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。<br><br>-g オプションを指定するかどうかを決定するときには、すべてのマテリアライズドビューを初期化すると再ロードプロセスの実行時間が大幅に長くなる可能性があることを考慮に入れてください。しかし、-g オプションを指定しない場合、初期化されていないマテリアライズドビューを最初に使用しようとしたクエリはビューの初期化が完了するまで待つことになり、予想外の遅延が生じることがあります。-g オプションを指定しない場合は、再ロードの完了後に、マテリアライズドビューを手動で初期化することもできます。「マテリアライズドビューの初期化」『SQL Anywhere サーバー SQL の使用法』を参照してください。<br><br>**テキストインデックス**　MANUAL REFRESH と定義されているテキストインデックスは、デフォルトでは再ロード後に初期化されません。このようなテキストインデックスを再ロードプロセス中に初期化するには、-g オプションを指定してください。-g を指定すると、データベースサーバーによって sa_refresh_text_indexes システムプロシージャーが呼び出されます。「sa_refresh_text_indexes システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |

| オプション | 説明 |
| --- | --- |
| **-ii** | UNLOAD 文を使用してデータベースからデータを抽出し、*reload.sql* ファイル内の LOAD 文を使用してデータベースにデータを再配置します。これはデフォルトです。 |
| **-ix** | UNLOAD 文を使用してデータベースからデータを抽出し、*reload.sql* ファイル内の Interactive SQL の INPUT 文を使用してデータベースにデータを再配置します。 |
| **-k** | sa_diagnostic_auxiliary_catalog テーブルにデータを配置します。このテーブルは、テーブル、ユーザー、プロシージャーなどのデータベースオブジェクト ID をソースデータベースからトレーシングデータベースにマッピングします。このとき、すべてのヒストグラムがアンロード／再ロードされます。このオプションは、診断トレーシング情報を受け取るトレーシングデータベースを作成する場合に使用します。さらに、このオプションによって、監査が運用データベースで有効になっている場合であっても、トレーシングデータベースでは監査を確実に無効にします。sa_diagnostic_auxiliary_catalog テーブルによって、サーバーはトレーシングデータを収集したときの状態を (インデックスコンサルタントやアプリケーションプロファイリングなどの目的で) シミュレートすることができます。このオプションが特に役立つのは、-n オプションと一緒に指定した場合です。「診断トレーシング」『SQL Anywhere サーバー SQL の使用法』と「sa_diagnostic_auxiliary_catalog テーブル」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **-kd** | データベースを 1 つの DB 領域ファイルに再ロードします。このオプションは、トレーシングデータベースを作成しているコンピューターのディレクトリ構造が運用データベースと同じでない場合に、トレーシングデータベースを作成するのに便利です。「外部トレーシングデータベースの作成 (Sybase Central の場合)」『SQL Anywhere サーバー SQL の使用法』を参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
|---|---|
| **-l** | データベースを再構築しても SYSTABCOL.max_identity の現在の値を維持します。<br><br>デフォルトでは、オートインクリメントカラムがあるテーブルを含むデータベースを再構築すると、データベースサーバーでは、テーブルの現在の内容に基づいて、各オートインクリメントカラムに使用できる次の値が計算されます。通常はこの処理で十分です。ただし、値の範囲の最後からローが削除されている場合は、値が再利用される可能性があり、この処理は適切ではありません。<br><br>-l オプションを指定すると、オートインクリメント値が含まれるテーブルごとに、sa_reset_identity システムプロシージャーへの呼び出しが生成される *reload.sql* スクリプトに追加され、SYSTABCOL.max_identity の現在の値が維持されます。<br><br>参照：<br><br>● 「AUTOINCREMENT カラムのあるテーブルの再ロード」『SQL Anywhere 12 変更点とアップグレード』<br>● 「SYSTABCOL システムビュー」『SQL Anywhere サーバー SQL リファレンス』<br>● 「sa_reset_identity システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』 |
| **-m** | レプリケーションに参加しているデータベースでユーザー ID を保存しません。 |
| **-n** | データベースのデータをアンロードしません。*reload.sql* には、データベースの構造だけを構築する SQL 文が記述されています。*reload.sql* ファイルに LOAD TABLE 文または INPUT 文を追加する場合は、-n ではなく -nl を使用してください。 |
| **-nl** | 構造体をアンロードします (-n オプションと同じ動作)。ただし、生成される *reload.sql* ファイルには、各テーブルに対する LOAD TABLE 文または INPUT 文も含まれます。このオプションを使用した場合、ユーザーデータはアンロードされません。-nl を指定するときには、LOAD/INPUT 文を生成できるようにデータディレクトリも指定する必要があります。ただし、このディレクトリにファイルは書き込まれません。このオプションを指定すると、データをアンロードしない再ロードスクリプトを生成できます。データをアンロードするには、-d を指定します。データベースにデータのアンロードが不要なテーブルがある場合は、dbunload -d -e table-name と指定することによって、そのテーブルのデータをアンロードの対象から除外することができます。 |

| オプション | 説明 |
|---|---|
| **-no** | データベースオブジェクトを名前順にアンロードします。デフォルトでは、dbunload は作成された順にオブジェクトを生成します。作成順序が異なる同一のオブジェクトが複数のデータベース内にある場合は、-no オプションを指定すると、データベーススキーマを比較する際に役立ちます。<br><br>-no を指定すると、オブジェクトの定義は *reload.sql* ファイル内でオブジェクトタイプのアルファベット順で分類されます。<br><br>● ユーザー<br>● グループメンバーシップ<br>● テーブル<br>● インデックスと外部キー<br>● ビュー<br>● プロシージャー<br>● ファンクション<br>● トリガー<br>● イベント<br>● Web サービス<br><br>オブジェクトの定義は、所有者と名前の順で出力されます。外部キー、役割名、トリガー名などの 3 つ目の要素が順序に含まれることがあります。<br><br>-no オプションは、-n、-nl、-ar、-an、-ac オプションと一緒には使用できません。同じバージョンのデータベースサーバーを使用して作成されたデータベースの再ロードスクリプトを比較するときは、比較を簡単にするために、-no オプションを使用することをおすすめします。これは、オブジェクトの定義間にわずかな違いがあるからです。<br><br>**警告**<br>オブジェクトの作成順序は重要な意味を持つため、生成されたファイルを使用して新しいデータベースを作成しないでください。たとえば、プロシージャー p2 がプロシージャー p1 を呼び出し、p1 が結果セットを返す場合、p1 を p2 の前に定義することが重要です。-no オプションを指定して生成された *reload.sql* スクリプトファイルを実行しようとすると、エラーが発生します。 |
| **-o** *filename* | 指定したファイルに、出力メッセージを書き込みます。このファイルのロケーションは dbunload に対する相対パスで指定します。 |
| **-p** *char* | 外部アンロード (dbunload -x オプション) で使用するデフォルトのエスケープ文字 (¥) を別の文字に変更できます。このオプションは、このユーティリティをコマンドプロンプトで実行する場合のみ使用できます。 |

| オプション | 説明 |
|---|---|
| **-q** | クワイエットモードで実行します (メッセージを表示しません)。このオプションは、このユーティリティをコマンドプロンプトで実行する場合のみ使用できます。-q を指定する場合、-y も指定する必要があります。-y を指定しなければ、*reload.sql* がすでに存在する場合、アンロードは失敗します。 |
| **-qc** | アンロードの完了後にメッセージウィンドウを閉じます。デフォルトでは、dbunload のメッセージウィンドウはユーザーが閉じるまで開いたままです。このオプションは Windows Mobile でのみ使用できます。 |
| **-qr** | テーブルのロードやインデックスの作成を行うときに、進行メッセージの作成と表示をできないようにします。 |
| **-r** *reload-file* | 生成される再ロード SQL スクリプトファイルの名前とディレクトリを変更します。デフォルトは現在のディレクトリの *reload.sql* です。ディレクトリは、サーバーではなく、クライアントアプリケーションの現在のディレクトリに相対します。 |
| **-t** *table, ...* | アンロードするテーブルのリストを指定します。デフォルトでは、すべてのテーブルがアンロードされます。-n オプションと一緒に使うと、テーブル定義のセットだけをアンロードできます。大文字と小文字を区別するデータベースであっても、テーブル名では常に大文字と小文字が区別されません。<br><br>-t オプションを使用して作成した *reload.sql* ファイルにはすべてのデータベーステーブルが含まれるわけではないので、このファイルを使用してデータベースを再構築しないでください。外部キーによって参照されているテーブルがある場合、そのテーブルの内容がなければ、データベースを再構築できません。<br><br>-t オプションを使用する場合は、必ず -d オプションも使用し、-t で指定したテーブルのデータをアンロードすることをおすすめします。 |
| **-u** | データの順序付けにインデックスを使用できないようにします。矛盾したインデックスを持つデータベースをアンロードする場合に、このオプションを使用すると、矛盾したインデックスがデータの順序付けに使われることはありません。通常、各テーブルでは、プライマリキーまたはクラスタードインデックスが定義されている場合、それに基づいてデータが順序付けられます。 |
| **-v** | アンロードしているテーブルの名前と、アンロードが完了したローの数を表示します。このオプションは、dbunload をコマンドプロンプトで実行する場合にのみ使用できます。 |
| **-xi** | dbunload クライアントへデータをアンロードし、生成された再ロードスクリプトファイル *reload.sql* 内の LOAD 文を使用することで外部アンロードを実行し、データベースにデータを再配置します。 |

| オプション | 説明 |
|---|---|
| **-xx** | dbunload クライアントへデータをアンロードし、生成された再ロードスクリプトファイル *reload.sql* 内の Interactive SQL の INPUT 文を使用することによって外部アンロードを実行し、データベースにデータを再配置します。 |
| **-y** | 確認メッセージを表示することなく、既存のスクリプトファイルを置き換えます。-q を指定する場合、-y も指定する必要があります。-y を指定しなければ、dbunload が既存のスクリプトファイルを検出した場合、アンロードは失敗します。<br><br>レプリケーションに参加しているデータベースをアンロードする場合、特殊な考慮事項がいくつかあります。「リモートデータベースの抽出」『SQL Remote』と「SQL Remote のアップグレード」『SQL Anywhere 12 変更点とアップグレード』を参照してください。 |
| *directory* | アンロードされたデータを格納するディレクトリを指定します。*reload.sql* スクリプトファイルは、常にユーザーの現在のフォルダーと相対的になります。 |

## 備考

**注意**
既存のデータベースをバージョン 12 のデータベースに再構築する方法については、「SQL Anywhere サーバーのアップグレード」『SQL Anywhere 12 変更点とアップグレード』を参照してください。

バージョン 10.0.0 以降のデータベースで dbunload を使用する場合は、使用する dbunload のバージョンが、データベースへのアクセスに使用するデータベースサーバーのバージョンと一致する必要があります。dbunload のバージョンがデータベースサーバーのバージョンより古いまたは新しい場合は、エラーがレポートされます。

アンロードユーティリティを使ってデータベースをアンロードし、指定したディレクトリの中にデータファイルのセットを入れることができます。アンロードユーティリティは、データベースを再構築するために、Interactive SQL スクリプトファイルを作成します。また、各テーブルのすべてのデータを、カンマ区切りの形式で、指定したディレクトリ内のファイルにアンロードします。バイナリデータはエスケープシーケンスを使って正しく再現できます。

内部アンロード／再ロードを実行すると、UPDATE ISYSUSER 文の発行により、各ユーザーの現在のステータスに関する情報がアンロードされます。外部アンロード／再ロードでは、この情報は含まれず、すべてのユーザーのステータスがリセットされます。「ログインポリシー」553 ページを参照してください。

データベースをアンロードしてから再ロードして再構築すると、再構築されたデータベースが元のデータベースより小さくなる場合があります。データベースのサイズが縮小されるのは、SQL Anywhere でのインデックス変更が原因と考えられます。問題やデータが失われたことを示すものではありません。

アンロードユーティリティ (dbunload) を使用してバージョン 11 以前のデータベースを再構築すると、元のデータベースでチェックサムがオフになっていても、新しいデータベースではデフォ

ルトでグローバルチェックサムが有効になります。おすすめできる方法ではありませんが、ALTER DATABASE 文を使用して、データベースのグローバルチェックサムを無効にすることができます。「ALTER DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』と「チェックサムを使用した破損の検出」1120 ページを参照してください。

**注意**
リカバリが必要なバージョン 9 以前のデータベースは、バージョン 10 以降のアンロードユーティリティ (dbunload) を使用して再ロードできません。このようなデータベースはバージョン 9 以前の dbunload を使用して再ロードしてください。

また、アンロードユーティリティを使って、既存のデータベースから直接新しいデータベースを作成できます。これにより、通常のディスクファイルに書き込まれたデータベースの内容に関するセキュリティ問題が生じる可能性を回避できます。

テーブルデータのみをアンロードする場合は、Sybase Central の **[データのアンロード]** ウィンドウを使用すると 1 つの手順で行えます。

詳細については、「[データのアンロード] ウィンドウを使用したデータのエクスポート」『SQL Anywhere サーバー SQL の使用法』を参照してください。

レプリケーションに参加しているデータベースをアンロードする場合、特殊な考慮事項がいくつかあります。「リモートデータベースの抽出」『SQL Remote』を参照してください。

次の方法で、アンロードユーティリティにアクセスできます。

- Sybase Central の**データベースアンロードウィザード**を使用する。「データベースアンロードウィザードを使用したデータのエクスポート」『SQL Anywhere サーバー SQL の使用法』を参照してください。

- コマンドプロンプトで、dbunload コマンドを入力する。バッチまたはコマンドファイルへの組み込みには、このユーティリティが便利です。

アンロードユーティリティは、DBA 権限のあるユーザー ID で実行してください。この方法でないと、すべてのデータをアンロードする権限を確実に持つことができません。また、*reload.sql* ファイルも DBA 権限を持つユーザーが実行する必要があります (通常、このファイルは、パスワードが sql である DBA を唯一のユーザー ID とする新しいデータベースで実行します)。

データベースサーバーの -gl オプションは、データベースからデータをアンロードするときに必要となるパーミッションを制御します。「-gl dbeng12/dbsrv12 サーバーオプション」236 ページを参照してください。

ユーザー ID dbo は、データベース内のシステムオブジェクトを所有します。これにはビューやストアドプロシージャーも含まれます。

アンロードユーティリティでは、データベース作成時に dbo ユーザー ID 用に作成されたオブジェクトをアンロードしません。データベースをアンロードすると、システムプロシージャーの再定義など、これらのオブジェクトに加えられた変更は失われます。データベースの初期化以降に dbo ユーザー ID によって作成されたオブジェクトは、アンロードユーティリティでアンロードされ、これらのオブジェクトは保存されます。

データベースをアンロードした場合、システムオブジェクトに対するパーミッションの変更はアンロードされません。必要に応じて、新しいデータベースでパーミッションの付与または取り消しを行う必要があります。

**ヒント**
データベースを再構築する前に、データを含めないでデータベースを再ロードして、再ロードプロセスを検証することをおすすめします。次のようなコマンドを実行してください。

```
dbunload -n -an new.db -c "UID=your-user-id;PWD=your-password;DBF=original-database-file"
```

元のデータベースに問題が見つかった場合は、修正してから再構築してください。

デフォルトモードの場合、または -ii か -ix を使用する場合、データを格納するために dbunload が使用するディレクトリは、ユーザーの現在のディレクトリではなく、データベースサーバーとの相対ディレクトリになります。

-xi または -xx を使用する場合、ディレクトリはユーザーの現在のフォルダーとの相対ディレクトリになります。

このモードでファイル名とパスを指定する方法の詳細については、「UNLOAD 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

テーブルのリストがない場合、すべてのデータベースをアンロードします。テーブルのリストがある場合、リストにあるテーブルだけをアンロードします。

アンロードされたデータには、*reload.sql* ファイルで生成された LOAD TABLE 文のカラムリストが含まれています。カラムリストのアンロードにより、テーブル内のカラムを並べ替えることができます。テーブルは削除または再作成することができ、その後 *reload.sql* を使用して再移植できます。

dbunload が生成する LOAD TABLE 文は、検査制約と計算カラムを無効にします。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』を参照してください。

**注意**
データベースを再構築した後は、データベースでマテリアライズドビューをリフレッシュすることをおすすめします。「手動マテリアライズドビューのリフレッシュ」『SQL Anywhere サーバー SQL の使用法』を参照してください。

**注意**
トレーシング情報は、データベースのアンロードまたは再ロード操作の一環としてアンロードされません。トレーシング情報を別のデータベースに転送する場合は、sa_diagnostic_* テーブルの内容をコピーして、手動で転送してください。ただし、この方法はおすすめしません。

### 内部アンロードと外部アンロード、内部再ロードと外部再ロード

-ii、-ix、-xi、-xx オプションを使用して、内部アンロード、外部アンロード、内部再ロード、外部再ロードを組み合わせます。内部文 (UNLOAD/LOAD) を使用すると、外部コマンド

(Interactive SQL の INPUT と OUTPUT 文) に比べ、パフォーマンスが大幅に向上します。ただし、内部文はサーバーから実行されるため、ファイルとディレクトリは、データベースサーバーに対する相対パスを使用します。外部コマンドを使用すると、ユーザーの現在のディレクトリに対する相対パスを使用することになります。

Sybase Central では、サーバーに相対してアンロードするか、クライアントに相対してアンロードするかを指定できます。「UNLOAD 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

データベースのアンロード、再ロード、または再構築を行うために外部アンロードや再ロードを行う場合、データベースの文字セットと dbunload を実行しているホストシステムの文字セットとの間に互換性がないと、文字セットを変換することでデータが破損する可能性があります。これは、データベースの文字セットとホストシステムの文字セットの間で変換が行われることが原因です。

この問題を回避するため、データベース用の接続文字列にデータベースの文字セットを指定します (-c オプションと -ac オプションを使用)。たとえば、データベースの文字セットが UTF-8 である場合、接続文字列に "charset=utf-8" を含めます。

```
dbunload -c UID=user-ID;PWD=password;
CHARSET=utf-8;DBF=filename -ac UID=user-ID;
PWD=password;CHARSET=utf-8;Host=host-name -xx
```

外部アンロードを行うと、*reload.sql* の先頭に、コメントアウトされた CREATE DATABASE 文が含まれます。この文を使用して、アンロードしているデータベースと同じデータベースを作成できます。

アンロードしたデータベースがバージョン 9 以前の SQL Anywhere で作成され、カスタム照合があった場合は、COLLATION 句は次のようになります。

```
COLLATION collation-label DEFINITION collation-definition
```

ここで *collation-definition* はカスタム照合を指定する文字列です。

カスタム照合を保持する唯一の方法は、データベースを 1 ステップ (内部アンロード) で再構築することです。データベースをアンロードしてから、作成したデータベースにスキーマとデータをロードする場合は、提供されるいずれかの照合を使用する必要があります。

アンロードしたデータベースが強力な暗号化を使用して作成されていた場合は、CREATE DATABASE 文の KEY 句の値が疑問符 3 つ (**???**) として表示されます。

## アンロードの失敗

テーブルデータの再ロードとテーブル上のインデックスの再構築が完了した後で、-ar または -an を使用してデータベースの内部再構築を行っているときに障害が発生した場合、dbunload は現在のディレクトリに *unprocessed.sql* というファイルを作成します。このファイルには、障害が発生したために実行されなかった文が記録され、障害の原因となった文もコメントとして示されています。次に、*unprocessed.sql* ファイルの例を示します。

```
-- The database reload failed with the following error:
-- ***** SQL error: the-SQL-error
-- This script contains the statements that were not executed as a
-- result of the failure. The statement that caused the failure is
```

```
-- commented out below. To complete the reload, correct the failing
-- statement, remove the surrounding comments and execute this script.
/*
the failing statement
go

*/

setuser "DBA"
go

... the remainder of the statements to be processed
```

このファイルがあると、障害の原因となった文を修正、削除、または変更できます。*unprocessed.sql* ファイルは、すべてのテーブルデータと参照整合性制約が再ロードされた後に作成されます。Interactive SQL を使用して新しいデータベースに接続し、更新された *unprocessed.sql* ファイルを実行できます。この方法で、再構築を最初からやり直さずにデータベースの再構築を完了でき、場合によっては時間を大幅に節約できます。

*unprocessed.sql* ファイルを生成するとき、dbunload は停止して失敗を示すエラーコードを返し、再構築が失敗したことを他のツールやスクリプトに通知します。

-ar を使用したデータベースの内部再構築で障害が発生すると、新しいデータベースファイルとトランザクションログファイルに *.db* と *.log* というファイル拡張子がそれぞれ付けられます。次の手順を使用して、*unprocessed.sql* ファイルを適用し、再ロードを手動で完了します。

1. 新しいデータベースを起動します。
2. 更新済みの *unprocessed.sql* ファイルを適用します。
3. データベースを停止します。
4. *original-name.db* と *original-name.log* を、新しいディレクトリに移動します。
5. *original-name.dbr* と *original-name.logr* ファイルの名前を *original-name.db* と *original-name.log* にそれぞれ変更します。
6. 次のコマンドを実行します。

   ```
   dblog -t original-name.log original-name.db
   ```

## 暗号化されたデータベース

テーブル暗号化が有効であるデータベースを再構築するとき、-er または -et を指定して、新しいデータベースのテーブルで暗号化を有効にするかどうかを指定する必要があります。このオプションを指定しないと、データを新しいデータベースにロードしようとしたときにエラーが発生します。

強力に暗号化されたデータベースをアンロードする場合は、暗号化キーを指定します。DatabaseKey (DBKEY) 接続パラメーターを使用して、コマンドに暗号化キーを指定できます。または、暗号化キーを読み取り可能な文字で入力するのではなく、暗号化キーを要求するプロンプトを表示させる場合は、次に示すように -ep サーバーオプションを使用できます。

```
dbunload -c "DBF=myenc.db;START=dbeng12 -ep"
```

-an オプションを使用してデータベースをアンロードし、新しいデータベースに再ロードするときに、-ek または -ep オプションを使用して新しいデータベースに対して暗号化キーを設定する場合は、次の点を考慮してください。

- 元のデータベースが強力に暗号化されている場合は、-ek または -ep オプションではなく、-c オプションで DatabaseKey (DBKEY) 接続パラメーターを使用して、元のデータベースに対するキーを指定する必要があります。

- -ek または- ep オプションを使用すると、暗号化されていないデータベースをアンロードし、強力に暗号化された新しいデータベースに再ロードできます。-ep および -an を使用するときは、キーが正しいことを確認してください。正しくない場合は、アンロードは失敗します。

- 元のデータベースが強力に暗号化されていても、-ek または -ep オプションを使用しないと、新しいデータベースは単純暗号化で暗号化されます。

- -an が指定されていない場合は、-ek と -ep オプションは無視されます。dbunload の -ek と -ep オプションが新しいデータベースに適用されるのに対し、データベースサーバー (dbeng12/dbsrv12) オプションと DBKEY= は既存のデータベースに適用されます。

- 同期またはレプリケーションに使用しているデータベースを再構築する場合、dbunload は、-ek または -ep オプションで指定された暗号化キーが元のデータベースおよび再構築されるデータベースの暗号化キーであると見なします。

暗号化の詳細については、「-ep dbeng12/dbsrv12 サーバーオプション」226 ページと「DatabaseKey (DBKEY) 接続パラメーター」335 ページを参照してください。

### データベースの再構築

データベースをアンロードするときには、まずそのデータベースが動作していないことを確認します。次に、dbunload を実行します。このとき、DBA ユーザーとパスワードを指定し、DBF= 接続パラメーターでデータベースを指定します。

データベースを再ロードするには、新しいデータベースを作成し、生成された *reload.sql* スクリプトファイルを Interactive SQL を介して実行します。

アンロードと再ロードを一度に行うには、dbunload を実行するときに、上記のアンロードに必要なオプションを指定するとともに、-an オプションで新しいデータベースファイルの名前を指定します。-ac と -an オプションの説明を参照してください。

## アップグレードユーティリティ (dbupgrad)

**注意**
アップグレードユーティリティ (dbupgrad) を使用して、SQL Anywhere 9.0.2 以前のデータベースを SQL Anywhere 12 にアップグレードすることはできません。SQL Anywhere 9.0.2 以前のデータベースを 12 にアップグレードするには、アンロードと再ロードを実行してデータベースを再構築する必要があります。「SQL Anywhere サーバーのアップグレード」『SQL Anywhere 12 変更点とアップグレード』を参照してください。

システムテーブルとビューを更新し、新しいデータベースオプションを追加し、すべてのシステムストアドプロシージャーを再作成します。また、jConnect サポートをインストールし、データベース内の Java のサポートを変更します。

## 構文

**dbupgrad** [ *options* ]

| オプション | 説明 |
|---|---|
| *@data* | 指定された環境変数または設定ファイルからオプションを読み込みます。「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル非表示ユーティリティを使用して、設定ファイルの内容を読みにくくすることができます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |
| **-c** *"keyword=value; ..."* | 接続パラメーターを指定します。「接続パラメーター」320 ページを参照してください。<br><br>ユーザー ID には DBA 権限が必要です。<br><br>たとえば、次のコマンドは、sample12 というデータベースにパスワード sql を指定してユーザー DBA として接続し、jConnect サポートをインストールせずにアップグレードします。<br><br>dbupgrad -c "UID=DBA;PWD=sql;DBF=c:¥sa12¥sample12.db" -i |
| **-i** | jConnect メタデータのサポートを除外します。jConnect JDBC ドライバーを使用してシステムカタログ情報にアクセスする場合は、このオプションを使用しないでください。このオプションを指定した場合でも、システムカタログ情報にアクセスしないかぎり、jConnect を使用できます。必要に応じて、Sybase Central または ALTER DATABASE UPGRADE 文を使用して、jConnect メタデータのサポートを後から追加することもできます。「jConnect システムオブジェクトのデータベースへのインストール」『SQL Anywhere サーバー プログラミング』と「ALTER DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| **-o** *filename* | 指定されたファイルに出力メッセージを書き込みます。 |
| **-q** | クワイエットモードで実行します (ウィンドウを表示しません)。 |

## 備考

**警告**
必ずデータベースファイルをバックアップしてからアップグレードしてください。既存のファイルにアップグレードを適用した場合、アップグレードに失敗すると、これらのファイルは使用できなくなります。「バックアップとデータリカバリ」1071 ページを参照してください。

dbupgrad ユーティリティは、このソフトウェアの以前のバージョンで作成されたデータベースをアップグレードし、現在のバージョンが提供する機能を使用できるようにします。アップグレードできる最も古いバージョンは SQL Anywhere 10.0.0 です。このソフトウェアの初期のリリースで作成したデータベースに対して、そのバージョン以降のデータベースサーバーを実行できますが、データベースを作成した後に導入された機能の中には、データベースをアップグレードしないかぎり使用できないものもあります。

アップグレードユーティリティを使用して、現在ミラーリングされているデータベースをアップグレードすると、エラーメッセージが返されます。

**注意**
データベースをアップグレードした後は、データベースでマテリアライズドビューをリフレッシュすることをおすすめします。「手動マテリアライズドビューのリフレッシュ」『SQL Anywhere サーバー SQL の使用法』を参照してください。

アップグレードユーティリティを使用すると、システムテーブルやビューの更新、新しいデータベースオプションの追加、データベースオプションのリストア、すべてのシステムストアドプロシージャーの再作成、jConnect サポートのインストールやデータベース内の Java のサポートの変更を実行できます。

**注意**
データベースファイルの物理的な再編成を必要とする機能は、dbupgrad を使用しても使用できるようにはなりません。そのような機能には、インデックスの拡張やデータの格納に関する変更が含まれています。これらの拡張機能を利用するには、データベースのアンロードと再ロードを行ってください。「SQL Anywhere サーバーのアップグレード」『SQL Anywhere 12 変更点とアップグレード』を参照してください。

SQL Anywhere での新バージョンやソフトウェアの更新が利用可能になったため、アップグレードユーティリティを使って新しい機能を利用できます。

データベースのアップグレードでは、データベースをアンロードして再ロードする必要はありません。

アップグレードされたデータベース上でレプリケーションを使用する場合は、トランザクションログをアーカイブし、アップグレードされたデータベース上で新しいログを起動してください。

次の方法で、アップグレードユーティリティにアクセスできます。

●Sybase Central の**データベースアップグレードウィザード**を使用する。

●Interactive SQL の ALTER DATABASE UPGRADE 文を使用する。

●コマンドプロンプトで、dbupgrad コマンドを入力する。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

**参照**

- 「ALTER DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』

# 検証ユーティリティ (dbvalid)

データベース内のテーブルとマテリアライズドビューの一部またはすべてについて、インデックスとキーを検証します。

**構文**

**dbvalid** [ *options* ] [ *object-name*, ... ]

| オプション | 説明 |
|---|---|
| *@data* | 指定された環境変数または設定ファイルからオプションを読み込みます。「設定ファイル」909 ページを参照してください。<br><br>設定ファイル内のパスワードなどの情報を保護する場合は、ファイル非表示ユーティリティを使用して、設定ファイルの内容を読みにくくすることができます。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| オプション | 説明 |
| --- | --- |
| **-c** *"keyword=value; ..."* | データベース接続パラメーターを指定します。 接続パラメーターの詳細については、「接続パラメーター」320 ページを参照してください。<br><br>DBA 権限または VALIDATE 権限のあるユーザー ID を使用してください。<br><br>たとえば、次のコマンドは、パスワード sql を指定してユーザー DBA として接続し、データベース (*c:¥salesdata.db* のすべてのテーブルとマテリアライズドビューを含む) を検証します。<br><br>dbvalid -c "UID=DBA;PWD=sql;DBF=c:¥salesdata.db" |
| **-d** | データベース内のすべてのページが正しいオブジェクトに属しているかどうかを検証し、チェックサム検証を実行します。-d オプションはインデックスの正当性を検証します。-d オプションを -i、-s、または -t オプションと一緒に使用することはできません。<br><br>「VALIDATE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| **-fx** | プライマリキー／外部キー参照整合性チェックを無効にします。 |
| **-i** | *object-name* をインデックスのリストとして定義します。 |
| **-o** *filename* | 指定したファイルに、出力メッセージを書き込みます。 |
| **-q** | 出力メッセージをクライアントに表示しません。ただし、-o オプションを使用してメッセージをファイルに書き込むことは可能です。 |

| オプション | 説明 |
| --- | --- |
| **-s** | チェックサムを使用してデータベースを検証します。チェックサムは、データベースページがディスク上で変更されたかどうかを判断するために使用します。チェックサム検証では、データベースの各ページをディスクから読み取り、チェックサムが存在すればそのチェックサムを計算します。計算されたチェックサムがページに保存されているチェックサムと異なる場合、ディスク上でページが変更されていることを意味し、エラーが返されます。無効なページのページ番号がデータベースサーバーメッセージウィンドウに表示されます。-s オプションは、-d、-i、-t、または -fx オプションと一緒に使用することはできません。 |
| **-t** | *object-name* をテーブルとマテリアライズドビューのリストとして定義します。 |
| *object-name* | 検証するテーブルまたはマテリアライズドビューの名前を指定します。<br><br>-i を使用した場合、*object-name* は検証するインデックスを表します。 |

**備考**

検証ユーティリティを使用すると、データベース内のテーブルとマテリアライズドビューの一部またはすべてについて、インデックスとキーを検証できます。検証ユーティリティでは、データベース内のすべてのページが正しいオブジェクトに属し、ページチェックサムが正しいことを確認するデータベースファイル構造を検証できます。デフォルトでは、dbvalid はデータベース内のすべてのテーブル、マテリアライズドビュー、インデックスを検証し、データベースファイル構造を検証します。

テーブルを検証するときに、dbvalid はすべてのテーブルのインデックスも検証して、テーブル内の一連のローと値が各インデックスに含まれる一連のローと値に一致するかどうかを確認します。テーブル内のすべての BLOB もトラバースされ、BLOB アロケーションマップが検証されて、孤立した BLOB が検出されます。また、検証ユーティリティは、すべてのインデックスページの物理構造、インデックスハッシュ値の順序、インデックスの一意性要件 (指定されている場合) も検査します。-fx オプションが指定されない場合、各外部キー値は、参照整合性制約がそのまま残っているかどうかについて、対応するプライマリキーテーブルで検索されます。

-i オプションが指定されたときには、dbvalid は object-list 内の各インデックスを検証します。インデックスの検証は、テーブルの検証とまったく同じように処理されますが、指定されたインデックスとその下にあるテーブルが検証されることだけが異なります。インデックスが外部キーの場合は、-fx オプションが指定されない限り、それぞれの値がプライマリキーテーブルで検索されます。

データベースクリーナーが実行している間にデータベースの検証を開始すると、データベースクリーナーの実行が終了するまで検証は実行されません。「sa_clean_database システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

検証ユーティリティを実行するためには、DBA 権限または VALIDATE 権限が必要です。

次の方法で、検証ユーティリティにアクセスすることもできます。

- Sybase Central の**データベース検証ウィザード**を使用する。「データベースの検証」1118 ページを参照してください。

- Interactive SQL から VALIDATE 文を使用する。「VALIDATE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

検証ユーティリティを通常のバックアップと一緒に使用すると、データベースのデータの整合性を保持できます。データベースのバックアップコピーを検証する場合は、バックアップのコピーを作成し、そのコピーを検証することをおすすめします。そうすることで、リカバリに使用されるファイルが変更されないようにすることができます。「バックアップとデータリカバリ」1071 ページを参照してください。

**警告**
データベースとトランザクションログのバックアップコピーには、どのような変更でも加えるべきではありません。バックアップ中に処理中のトランザクションがなかった場合、または BACKUP DATABASE WITH CHECKPOINT LOG RECOVER か WITH CHECKPOINT LOG NO COPY を指定した場合は、読み込み専用モードを使用するか、バックアップデータベースのコピーを検証して、バックアップデータベースの妥当性をチェックできます。

一方、トランザクションの処理中だった場合、または BACKUP DATABASE WITH CHECKPOINT LOG COPY を指定した場合は、検証の開始時にデータベースサーバーがデータベースのリカバリを実行する必要が生じます。リカバリを実行するとバックアップコピーに変更が加えられるため、元のデータベースからの以降のトランザクショログファイルが適用されなくなります。

検証ユーティリティの実行によってデータベースを自動的に起動する場合、データベースは読み込み専用モードで起動されます。この処理が、バックアップまたはリカバリのプランの一環として検証を行う場合に、データベースに変更が加えられることを防いでいます。

検証ユーティリティが、読み込み専用モードで起動していない実行中のデータベースに接続すると、警告が表示されます。この警告は、検証しようとしているデータベースを、リカバリプランの中で使用できないことを知らせるものです。バックアップの実行によって、dbbackup で作成したほとんどのデータベースには、リカバリが必要であることを示すマークが付けられます。検証対象のデータベースがリカバリを必要としており、そのデータベースを読み込み／書き込みモードで起動したい場合は、dbvalid の実行前にデータベースを起動するか、DBS 接続パラメーターに有効な値を指定してください。「DatabaseSwitches (DBS) 接続パラメーター」337 ページを参照してください。

次のコマンドはどちらも、*mycopy.db* データベースのリカバリが必要な場合に dbvalid を実行できるようにします。

```
dbvalid -c "UID=DBA;PWD=sql;DBF=mycopy.db;DBS=-n mycopy"
```

```
dbvalid -c "UID=DBA;PWD=sql;DBF=mycopy.db;DBS=-dh"
```

**警告**
テーブルまたはデータベース全体の検証は、データベースに変更を加えている接続がない場合に実行してください。そうしないと、実際に破損していなくても、何らかの形でデータベースが破損したことを示すエラーがレポートされます。

検証ユーティリティは、グローバルチェックサムが有効になっていないデータベースで、チェックサム違反に関する警告を返すことがあります。これは、データベースサーバーが、チェックサムが有効かどうかに関係なく、重要なデータベースページのチェックサムを自動的に計算するからです。データベースの一部のページにチェックサムが存在する場合もあります。これは、データベースが書き込みチェックサムを有効にして起動されたためです。

また、データベースサーバーは、Windows Mobile 上で実行されているデータベースや、信頼性の低いストレージメディア (リムーバブルドライブなど) 上で実行されているデータベースに対しても、自動的にチェックサムを作成します。「チェックサムを使用した破損の検出」1120 ページを参照してください。

検証は、検証対象のテーブルに対する排他アクセスを取得せず、検証中にデータベースが変更された場合には、失敗をレポートします。このことを考慮すれば、データベース上に他のアクティビティがない場合に、データベースの検証を行うのが最適です。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

検証中に行われる個々のチェックの詳細については、「VALIDATE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

### 参照

- 「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバー プログラミング』

## バージョン診断ユーティリティ (dbversion)

指定された実行プログラムに関する情報を返します。

### 構文

**dbversion** *executable-name*

### 備考

このユーティリティは、UNIX でのみ使用でき、SQL Anywhere の実行プログラムに関する情報を返します。

### 参照

- 「-v dbeng12/dbsrv12 サーバーオプション」287 ページ

### 例

次にコマンド例を示します。

```
$ dbversion /opt/sqlanywhere12/bin32/dbversion
```

このコマンドは、次のように dbversion 実行プログラムに関する情報を返します。

```
SQL Anywhere バージョン診断ユーティリティバージョン 12.0.1.3148
/opt/sqlanywhere12/bin32/dbversion: dbversion GA 12 0 1 3149 linux 2011/01/14 15:32:50 nothr 32
production
```

| フィールド | 説明 |
|---|---|
| dbversion | 実行プログラム名を返します。 |
| xx | インストールの種類を示す 2 文字のコードを返します。 |
| 12 | メジャーバージョン番号を返します。 |
| 0 | マイナーバージョン番号を返します。 |
| 0 | パッチ番号を返します。 |
| 1238 | ビルド番号を返します。 |
| linux | オペレーティングシステムコードを返します。 |
| 2008/04/02 23:31:54 | ビルド日時のスタンプを返します。 |
| nothr | スレッドモデルを返します (nothr または posix)。 |
| 32 | 実行プログラムのビット数を返します (32 または 64)。 |
| production | production または debug を返します。 |

# データベースのメンテナンス

この項では、データベースファイルのバックアップ方法と、イベントやスケジュールを使用してデータベース管理を自動化する方法について説明します。

## バックアップとデータリカバリ

「バックアップ」とはデータベース内の情報の全部または一部のコピーであり、物理的に別の場所に格納されます。データベースを使用できなくなった場合は、バックアップから「リストア」できます。バックアップを使用して、データベースが使用できなくなった時点までにコミットされたすべての変更をリストアできます。

実行中のデータベースをバックアップすると、他のユーザーによってデータベースが変更されている途中であっても、データが一貫した状態にあるデータベースのスナップショットを得ることができます。

オペレーティングシステムまたはデータベースサーバーで障害が発生するか、データベースサーバーが正常に停止されなかった場合は、データベースをリカバリする必要があります。データベースの起動時に、データベースサーバーは直前のセッションの最後にデータベースが正しく停止されたかどうかをチェックします。正しく停止されていない場合、データベースサーバーは自動リカバリ処理を実行して、最後にコミットされたトランザクションまでのすべての変更をリストアします。

**参照**


- 「バックアップの種類」1072 ページ
- 「データベースのバックアップ」1071 ページ
- 「バックアップユーティリティ (dbbackup)」913 ページ
- 「BACKUP 文」『SQL Anywhere サーバー SQL リファレンス』
- 「バックアップとリカバリのプランの設計」1101 ページ
- 「データベースのリカバリ」1089 ページ


## データベースのバックアップ

**♦ サーバー側バックアップを作成するには、次の手順に従います。**

バックアップを作成するときは、バックアップファイルをデータベースサーバーのコンピューターに保存するか、クライアントコンピューターに保存するかを決めます。

1. BACKUP 権限または REMOTE DBA 権限を持つユーザーでデータベースに接続します。

2. BACKUP DATABASE 文を実行します。次に例を示します。

```
BACKUP DATABASE DIRECTORY 'd:¥¥temp¥¥backup';
```

この文では、データベースファイルのバックアップコピーが、サーバーコンピューターの *d:¥temp¥backup* ディレクトリに作成されます。

また、-s オプションを指定して dbbackup を実行することで、サーバー側バックアップを作成することもできます。次に例を示します。

```
dbbackup -s -c "Host=sample_host;SERVER=myserver;DBN=demo;UID=DBA;PWD=sql"
"c:¥SQLAnybackup"
```

**♦ クライアント側バックアップを作成するには、次の手順に従います。**

1. BACKUP 権限または REMOTE DBA 権限を持つユーザーでデータベースに接続します。
2. クライアントコンピューターで、-c オプションを指定してバックアップユーティリティ (dbbackup) を実行します。次に例を示します。

```
dbbackup -c "Host=sample_host;SERVER=myserver;DBN=demo;UID=DBA;PWD=sql" "c:
¥SQLAnybackup"
```

**参照**


- 「バックアップユーティリティ (dbbackup)」913 ページ
- 「BACKUP 文」『SQL Anywhere サーバー SQL リファレンス』
- 「バックアップの種類」1072 ページ
- 「バックアップとリカバリのプランの設計」1101 ページ
- 「データベースのリカバリ」1089 ページ


# バックアップの種類

「オフライン」バックアップはデータベースファイルのコピーです。データベースが稼働していないときは、データベースファイルをコピーして「オフライン」バックアップを作成できます。オフラインバックアップは、データベースが稼働中でなく、データベースサーバーが正しく停止された場合にだけ実行するようにしてください。

「オンライン」バックアップは稼働中のデータベースに対して実行されます。実行中のデータベースをバックアップすると、他のユーザーによってデータベースが変更されている途中であっても、データが一貫した状態にあるデータベースのスナップショットを得ることができます。データベースのオンラインバックアップを作成するには、バックアップ権限または REMOTE DBA 権限が必要です。

次の表は、SQL Anywhere でサポートされているオンラインバックアップの種類を示したものです。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| バックアップの種類 | 説明 | その他の情報 |
| --- | --- | --- |
| フル | フルバックアップでは、データベースファイルとトランザクションログをバックアップします。通常、フルバックアップとフルバックアップの間にインクリメンタルバックアップを複数回実行します。 | 「フルバックアップ」 |
| インクリメンタル | トランザクションログのみのバックアップです。インクリメンタルバックアップを実行する前に、フルバックアップを実行する必要があります。 | 「インクリメンタルバックアップ」 |
| ライブ | 継続的なバックアップは、データベースの稼働中に実行するデータベースのバックアップです。 | 「ライブバックアップ」 |
| アーカイブ | メインデータベースファイル、トランザクションログ、すべての追加 DB 領域など、バックアップに必要なすべての情報が含まれた 1 つまたは複数のファイルの集合です。 | 「アーカイブバックアップ」 |
| イメージ | データベースファイルとトランザクションログ (任意) のコピーがそれぞれ別のファイルとして作成されます。Windows Mobile では、イメージバックアップのみが許可されています。 | 「イメージバックアップ」 |
| サーバー側 | データベースサーバーのコンピューターに作成されるバックアップです。 | 「サーバー側のバックアップ」 |
| クライアント側 | クライアントコンピューターに作成されるバックアップです。 | 「クライアント側バックアップ」 |

# フルバックアップ

**♦ フルバックアップを作成するには、次の手順に従います (概要)。**

「フルバックアップ」では、データベースファイルとトランザクションログの両方をバックアップします。

フルバックアップを実行するには、バックアップ権限または REMOTE DBA 権限が必要です。

1. BACKUP 権限または REMOTE DBA 権限を持つユーザーでデータベースに接続します。
2. 妥当性検査を実行して、データベースが壊れていないことを確認します。妥当性検査には、検証ユーティリティまたは sa_validate ストアドプロシージャーを使用します。
3. データベースファイルとトランザクションログのバックアップをとります。

> **注意**
> データベースファイルとトランザクションログファイルのバックアップコピーには、オンラインバージョンのファイルと同じ名前が付けられます。たとえば、サンプルデータベースのバックアップを作成する場合、バックアップコピーは *demo.db* および *demo.log* という名前になります。バックアップ文を繰り返す場合は、バックアップコピーを上書きしないように新しいバックアップディレクトリを選択してください。

**参照**


- 「データベースの検証」1118 ページ
- 「サーバー側のバックアップ」1079 ページ
- 「クライアント側バックアップ」1085 ページ
- 「バックアップを作成し、元のトランザクションログを削除する」1110 ページ
- 「バックアップを作成し、元のトランザクションログの名前を変更する」1108 ページ
- 「インクリメンタルバックアップ」1074 ページ
- 「バックアップの種類」1072 ページ


# インクリメンタルバックアップ

「「インクリメンタルバックアップ」」は、オンラインバックアップのタイプです。インクリメンタルバックアップを実行する前にフルバックアップを実行している場合、インクリメンタルバックアップはトランザクションログだけをバックアップし、リカバリでのみ使用できます。フルバックアップとフルバックアップの間にインクリメンタルバックアップを複数回実行可能です。「フルバックアップ」1074 ページを参照してください。

**♦ インクリメンタルバックアップを作成するには、次の手順に従います (概要)。**

1. BACKUP 権限または REMOTE DBA 権限を持つユーザーでデータベースに接続します。
2. 妥当性検査を実行して、データベースが壊れていないことを確認します。妥当性検査には、検証ユーティリティまたは sa_validate ストアドプロシージャーを使用します。「データベースの検証」1118 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

3. データベースファイルではなく、トランザクションログのバックアップを作成します。

> **注意**
> データベースファイルとトランザクションログファイルのバックアップコピーには、オンラインバージョンのファイルと同じ名前が付けられます。たとえば、サンプルデータベースのバックアップを作成する場合、バックアップコピーは *demo.db* および *demo.log* という名前になります。バックアップ文を繰り返す場合は、バックアップコピーを上書きしないように新しいバックアップディレクトリを選択してください。
>
> トランザクションログのバックアップコピーの名前を変更して、繰り返し可能なインクリメンタルバックアップコマンドを作成する方法については、「トランザクションログバックアップコピーに名前を付ける」1109 ページを参照してください。

**参照**


- 「サーバー側のバックアップ」1079 ページ
- 「クライアント側バックアップ」1085 ページ
- 「バックアップを作成し、元のトランザクションログを削除する」1110 ページ
- 「バックアップを作成し、元のトランザクションログの名前を変更する」1108 ページ
- 「フルバックアップ」1074 ページ
- 「バックアップの種類」1072 ページ


## ライブバックアップ

「ライブバックアップ」は、コンピューター全体に及ぶ障害からデータベースを保護するのに役立つ継続的なバックアップです。トランザクションログの冗長コピーを使用して、セカンダリコンピューターでシステムを再起動できます。

システム障害が発生した場合は、バックアップされたトランザクションログを使って、システムをすばやく再起動できます。しかし、データベースサーバーが処理しているロード量によっては、ライブバックアップは処理が遅れ、コミットされたすべてのトランザクションがバックアップされないことがあります。

通常、dbbackup ユーティリティはセカンダリコンピューターから実行してください。プライマリコンピューターを使用できなくなった場合、セカンダリコンピューターを使用してデータベースを再起動できます。データベースファイルとトランザクションログが、データベースを再起動するのに必要な情報を保持しています。

ライブバックアップの代わりにデータベースミラーリングを使用することもできます。「ライブバックアップとトランザクションログミラーの違い」1076 ページを参照してください。

### ライブバックアップと定期的なバックアップ

トランザクションログのライブバックアップの長さは、アクティブなトランザクションログと常に同じか短くなります。ライブバックアップの実行中に、別のバックアップがトランザクションログを再起動すると (dbbackup -r または dbbackup -x)、ライブバックアップは自動的にライブバックアップログをトランケートして、新しいトランザクションログの最初からライブバックアップを再起動します。

**参照**


- 「ライブバックアップの作成」1086 ページ
- 「ライブバックアップからの再起動」1093 ページ
- 「バックアップの種類」1072 ページ


## ライブバックアップとトランザクションログミラーの違い

ライブバックアップとトランザクションログミラーは両方とも、トランザクションログのセカンダリコピーを作成します。しかし、ライブバックアップの使用とトランザクションログミラーの使用には、次のようないくつかの違いがあります。

- **通常、ライブバックアップは別のコンピューターで実行される**　別のコンピューターでバックアップユーティリティを実行すると、バックアップログファイルへの書き込みはデータベースサーバーによっては行われません。また、データ転送は SQL Anywhere クライアント／サーバー通信システムによって実行されます。したがって、パフォーマンスへの影響を低減でき、信頼性も向上します。

  トランザクションログミラーの場合、別のコンピューターで実行することはおすすめしません。実行するとパフォーマンスに問題が発生したりデータが破壊されたりする可能性があり、コンピューター間の接続に障害が発生すると、データベースサーバーが停止します。

- **ライブバックアップは、コンピューターが使用できなくなる状況から保護する**　トランザクションログミラーを別のデバイスに保存しても、コンピューター全体が使用できなくなってしまえば、リカバリには時間がかかります。2 台のコンピューターで一連のディスクへのアクセスを共有するように構成するのも 1 つの方法です。

- **ライブバックアップはデータベースサーバーより処理が遅れることがある**　トランザクションログミラーには、コミットされたトランザクションを完全にリカバリするのに必要な情報がすべて含まれます。データベースサーバーが処理しているロード量によっては、ライブバックアップはトランザクションログミラーよりも処理が遅く、コミットされたトランザクションがすべてバックアップされないことがあります。

**参照**


- 「トランザクションログミラー」128 ページ


# バックアップフォーマット

「アーカイブバックアップ」では、データベースファイルとトランザクションログを 1 つまたは複数のファイル (通常はテープドライブ上) にコピーします。アーカイブバックアップは、サーバー側バックアップとしてのみ実行できます。

テープに直接バックアップする場合は、アーカイブバックアップを使用してください。それ以外の場合は、イメージバックアップの方がリストアが簡単なので、イメージバックアップを使用してください。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

「イメージバックアップ」では、データベースファイルとトランザクションログ (任意) のコピーをそれぞれ別のファイルとして作成します。

## アーカイブバックアップ

アーカイブバックアップは、メインデータベースファイル、トランザクションログ、すべての追加 DB 領域など、バックアップに必要なすべての情報が含まれた 1 つまたは複数のファイルの集合です。アーカイブバックアップは、サーバー側バックアップとしてのみ実行できます。アーカイブバックアップは、ファイルまたはテープドライブに保存できます。

アーカイブバックアップを作成するときは、作成されるファイルごとに、BACKUP 文で指定するファイル名に拡張子が追加されます (*.1*、*.2*、*.3*、以下同様)。

アーカイブバックアップでは、デフォルトで空きページがスキップされるため、バックアップが小規模で高速になる可能性があります。トランザクションログファイルには空きページが含まれないため、空きページを除去してもトランザクションログファイルのバックアップには影響がありません。したがって、大規模なトランザクションログファイルがあるデータベースで空きページを除去しても、トランザクションログファイルの規模が小さいデータベースほど利点はありません。

### 作成で使用するもの

アーカイブバックアップの作成には、BACKUP DATABASE 文または Sybase Central の**データベースバックアップウィザード**を使用します。

### 適用対象

アーカイブバックアップは、Windows Mobile 以外の Windows と UNIX の各プラットフォームでのみサポートされています。アーカイブバックアップは、サーバー側バックアップとしてのみ実行できます。

### リストアで使用するもの

アーカイブバックアップからのデータベースのリストアには、Sybase Central の**データベースリストアウィザード**または RESTORE DATABASE 文を使用します。

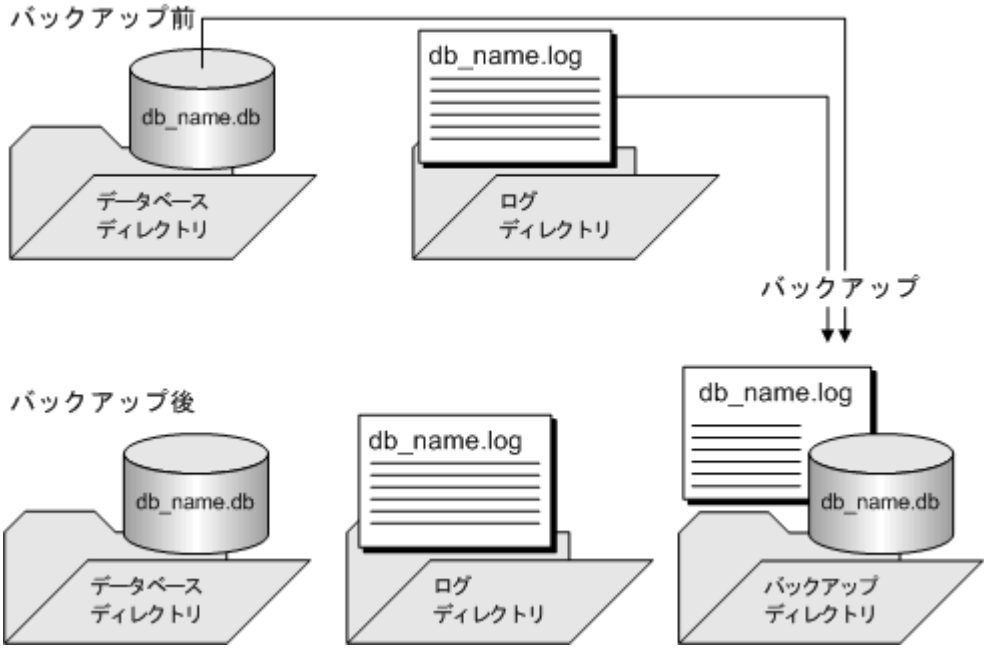
### 参照


- 「イメージバックアップ」1077 ページ
- 「データベースバックアップウィザードを使用した、アーカイブバックアップの作成」1083 ページ
- 「BACKUP DATABASE 文を使用したサーバー側のバックアップの実行」1081 ページ
- 「バックアップの種類」1072 ページ


## イメージバックアップ

イメージバックアップでは、データベースファイルとトランザクションログ (任意) のコピーがそれぞれ別のファイルとして作成されます。

最も単純なバックアップの種類は、データベースファイルとトランザクションログ (任意) のコピーをそれぞれ別のファイルとして作成するイメージバックアップです。イメージバックアップでは、データベースファイルとトランザクションログのコピーが作成され、トランザクションログはトランケートまたは置換されずそのまま残されます。すべてのバックアップにおいて、データベースファイルは所定の場所に格納されたままです。この種類のフルバックアップを次の図で説明します。




### 作成で使用するもの

イメージバックアップの作成には、バックアップユーティリティ (dbbackup)、**バックアップイメージ作成ウィザード**、または BACKUP DATABASE 文を使用します。

### 適用対象

イメージバックアップは、サポートされるすべてのプラットフォームで使用できます。Windows Mobile でサポートされているバックアップの種類は、イメージバックアップのみです。

テープにバックアップする場合は、アーカイブバックアップを使用します。

### 参照

- 「BACKUP 文」『SQL Anywhere サーバー SQL リファレンス』
- 「バックアップユーティリティ (dbbackup)」913 ページ
- 「BACKUP DATABASE 文を使用したサーバー側のバックアップの実行」1081 ページ
- 「イメージバックアップを作成する (バックアップイメージ作成ウィザードの場合)」1083 ページ
- 「バックアップの種類」1072 ページ
- 「アーカイブバックアップ」1077 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

# バックアップとリカバリの制限

データベースサーバーでは、バックアップ処理中に次のオペレーションを実行できないようにします。

●別のバックアップ (ライブバックアップを除く)。

●チェックポイント (バックアップ命令によって発生するチェックポイントは除く)。

●チェックポイントを発生させる文。「次の文は、暗黙的または明示的なチェックポイントを開始します。」132 ページを参照してください。

リカバリ (バックアップのリカバリを含む) の実行中は、データベースの別のユーザーによるいかなるアクションも許可されません。

**参照**


- 「次の文は、暗黙的または明示的なチェックポイントを開始します。」132 ページ
- 「チェックポイントログ」133 ページ
- 「バックアップコンポーネント」1113 ページ
- 「データベースサーバーがチェックポイントのタイミングを決定する方法」1114 ページ


# サーバー側のバックアップ

一般に、データベースサーバーのコンピューターにバックアップを作成する方が、クライアントコンピューターへのバックアップよりも高速です。これは、クライアント／サーバー通信システムを介してデータを転送する必要がないからです。サーバー側バックアップをアプリケーションに組み込むには、SQL 文を使用します。サーバー側バックアップは、次の方法で作成できます。

| ツール | その他の情報 |
|---|---|
| BACKUP 文 | BACKUP DATABASE 文は、物理デバイスレベルの並列処理を使用して、バックアップ操作の完了に必要な時間を節約します。並列バックアップは、Windows Mobile ではサポートされていません。「並列データベースバックアップ」1116 ページを参照してください。<br><br>「BACKUP DATABASE 文を使用したサーバー側のバックアップの実行」1081 ページを参照してください。 |

| ツール | その他の情報 |
| --- | --- |
| バックアップユーティリティ (dbbackup) | バックアップユーティリティ (dbbackup) は、物理デバイスレベルの並列処理を使用して、バックアップ操作の完了に必要な時間を節約します。並列バックアップは、Windows Mobile ではサポートされていません。「並列データベースバックアップ」1116 ページを参照してください。<br><br>「バックアップユーティリティ (dbbackup) を使用した、サーバー側のバックアップの実行」1082 ページを参照してください。 |
| データベースバックアップウィザード | このウィザードでは、アーカイブバックアップが作成されます。バックアップを保存するファイル名またはテープドライブを指定できます。「データベースバックアップウィザードを使用した、アーカイブバックアップの作成」1083 ページを参照してください。 |
| バックアップイメージ作成ウィザード | このウィザードでは、データベースの稼働中に各データベースファイルのコピーが作成されます。リカバリするには、すべてのファイルをデータベースサーバーのコンピューターの元のロケーションにコピーします。「イメージバックアップを作成する (バックアップイメージ作成ウィザードの場合)」1083 ページを参照してください。 |
| メンテナンスプラン作成ウィザード | このウィザードでは、データベースのバックアップを含むさまざまなタスクのスケジュールを作成できます。アーカイブ、ファイルイメージ、またはインクリメンタルバックアップを作成することを選択できます。「メンテナンスプランの作成」1104 ページを参照してください。 |
| DBBackup 関数 | DBBackup メソッド [データベースツール]『SQL Anywhere サーバー プログラミング』を参照してください。 |
| SQL Anywhere ボリュームシャドウコピーサービス (dbvss) | 「SQL Anywhere ボリュームシャドウコピーサービス (VSS)」1084 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**参照**


- 「クライアント側バックアップ」1085 ページ
- 「バックアップの種類」1072 ページ


## BACKUP DATABASE 文を使用したサーバー側のバックアップの実行

**♦ イメージバックアップを作成するには、次の手順に従います (SQL の場合)。**

次のバックアップでは、トランザクションログは変更されません。バックアップ作成時のその他のトランザクションログ管理オプションについては、「BACKUP 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

1. BACKUP 権限または REMOTE DBA 権限を持つユーザーでデータベースに接続します。
2. 次の文を実行して、イメージバックアップを作成します。

   ```
   BACKUP DATABASE
   DIRECTORY directory-name;
   ```

   BACKUP 文では、テキストファイル *backup.syb* にエントリが作成されます。*backup.syb* ファイルのロケーションの詳細については、「SALOGDIR 環境変数」450 ページを参照してください。

   イメージバックアップのリカバリについては、「イメージバックアップからのリストア」1091 ページを参照してください。

**♦ アーカイブバックアップをテープに作成するには、次の手順に従います (SQL の場合)。**

次のバックアップでは、トランザクションログは変更されません。バックアップ作成時のその他のトランザクションログ管理オプションについては、「BACKUP 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

1. BACKUP 権限または REMOTE DBA 権限を持つユーザーでデータベースに接続します。
2. 次の BACKUP 文を実行して、アーカイブバックアップをテープに作成します。

   ```
   BACKUP DATABASE
   TO archive-root
   [ ATTENDED { ON | OFF } ]
   [ WITH COMMENT comment-string ];
   ```

   ATTENDED オプションを OFF に設定すると、テープやディスク領域が十分でない場合、バックアップは失敗します。ATTENDED を ON に設定すると、バックアップアーカイブデバイス上の空き領域がなくなった場合、ユーザーによる対応を求められます。

   BACKUP 文では、テキストファイル *backup.syb* にエントリが作成されます。*backup.syb* ファイルのロケーションの詳細については、「SALOGDIR 環境変数」450 ページを参照してください。

アーカイブバックアップのリカバリについては、「アーカイブバックアップからのリストア」1091 ページを参照してください。

**参照**

- 「BACKUP 文」『SQL Anywhere サーバー SQL リファレンス』
- 「アーカイブバックアップ」1077 ページ
- 「イメージバックアップ」1077 ページ
- 「データベースのリカバリ」1089 ページ

**例**

次の文では、現在のデータベースとトランザクションログのイメージバックアップを作成し、それぞれ別のファイルに保存し、既存のトランザクションログ名を変更します。

```
BACKUP DATABASE
DIRECTORY 'd:¥¥temp¥¥backup'
TRANSACTION LOG RENAME;
```

次の文では、Windows コンピューター上で最初のテープドライブにアーカイブバックアップを作成します。

```
BACKUP DATABASE
TO '¥¥¥.¥¥tape0'
ATTENDED OFF
WITH COMMENT 'May 6 backup';
```

Windows の場合、最初のテープドライブは ¥¥.¥tape0 になります。円記号は、SQL 文字列のエスケープ文字であるため、各円記号は 2 つ重ねる必要があります。

## バックアップユーティリティ (dbbackup) を使用した、サーバー側のバックアップの実行

**♦ バックアップを作成し、元のトランザクションログを継続して使用するには、次の手順に従います (コマンドラインの場合)。**

dbbackup ユーティリティでは、データベースファイルとトランザクションログ (任意) のコピーをそれぞれ別のファイルとして含むイメージバックアップが作成されます。

ここでは、トランザクションログを変更しないバックアップについて説明します。バックアップ作成時のその他のトランザクションログ管理オプションについては、「バックアップユーティリティ (dbbackup)」913 ページを参照してください。

1. BACKUP 権限または REMOTE DBA 権限を持つユーザーでデータベースに接続します。

2. dbbackup ユーティリティを使用する場合は、次の構文を使用します。

   ```
   dbbackup -s -c "connection-string" [ -t ] backup-directory
   ```

   -s オプションを使用すると、データベースサーバー上にイメージバックアップが作成されます。

インクリメンタルバックアップを作成する場合だけ -t オプションを使用します。

**参照**


- 「インクリメンタルバックアップ」1074 ページ
- 「イメージバックアップ」1077 ページ
- 「イメージバックアップからのリストア」1091 ページ
- 「データベースのリカバリ」1089 ページ


**例**

次の例では、データベースサーバーのコンピューターの *c:¥SQLAnybackup* ディレクトリにイメージバックアップを作成します。

```
dbbackup -s -c "HOST=myhost;DBN=demo;UID=DBA;PWD=sql" "c:¥SQLAnybackup"
```

## データベースバックアップウィザードを使用した、アーカイブバックアップの作成

**♦ アーカイブバックアップを作成するには、次の手順に従います (Sybase Central のデータベースバックアップウィザードの場合)。**

**データベースバックアップウィザード**では、アーカイブバックアップが作成されます。Sybase Central でアーカイブバックアップを作成するときは、データベースをテープまたはディスクに直接バックアップするオプションがあります。

1. BACKUP 権限または REMOTE DBA 権限を持つユーザーでデータベースに接続します。
2. データベースを右クリックし、**[データベースのバックアップ]** をクリックします。
3. ウィザードの指示に従います。

**参照**


- 「アーカイブバックアップ」1077 ページ
- 「アーカイブバックアップからのリストア」1091 ページ
- 「データベースのリカバリ」1089 ページ


## イメージバックアップを作成する (バックアップイメージ作成ウィザードの場合)

**♦ バックアップを作成するには、次の手順に従います (Sybase Central のバックアップイメージ作成ウィザードの場合)。**

**[バックアップイメージ作成ウィザード]** を使用して、それぞれのデータベースファイルをコピーするバックアップを作成しますが、トランザクションログは変更されず、既存のトランザクションログを使用して作業を続けることができます。

1. バックアップ権限または REMOTE DBA 権限を持つユーザーでデータベースに接続します。

2. データベースを右クリックし、**[バックアップイメージの作成]** をクリックします。

> **ヒント**
> Sybase Central では、次の方法でデータベースの**バックアップイメージ作成ウィザード**にアクセスすることもできます。
>
> ●データベースを選択し、**[ファイル]** » **[バックアップイメージの作成]** をクリックする。
>
> ●**[ツール]** » **[SQL Anywhere 12]** » **[バックアップイメージの作成]** をクリックする。

3. **[次へ]** をクリックします。
4. **[バックアップするデータベースを指定してください。]** リストでデータベースを選択し、**[次へ]** をクリックします。
5. **[バックアップイメージを次のディレクトリに保存]** フィールドに、バックアップコピーを保存するディレクトリの名前を入力します。
6. **[バックアップするファイルを指定してください。]** リストでオプションを選択し、**[次へ]** をクリックします。
7. **[トランザクションログの処理方法を指定してください。]** リストで、**[同じトランザクションログを継続して使用]** をクリックします。
8. **[次へ]** をクリックします。
9. **[完了]** をクリックします。
10. リカバリするには、すべてのファイルをデータベースサーバーのコンピューターの元のロケーションにコピーします。

**参照**


- 「イメージバックアップ」1077 ページ
- 「イメージバックアップからのリストア」1091 ページ
- 「データベースのリカバリ」1089 ページ


## SQL Anywhere ボリュームシャドウコピーサービス (VSS)

SQL Anywhere は、Microsoft ボリュームシャドウコピーサービス (VSS) と互換性があります。VSS を使用すると、ディスクボリューム全体またはボリュームセットのポイントインタイムスナップショットを作成したり、SQL Anywhere データベースサーバーなどのアプリケーションで排他的に使用するために開かれているファイルのコピーを作成することができます。VSS は、32 ビット版の Microsoft Windows XP オペレーティングシステムと、32 ビットおよび 64 ビット版の Microsoft Windows 2003 以降のオペレーティングシステムでサポートされています。

デフォルトでは、SQL Anywhere VSS ライター (*dbvss12.exe*) が実行されている場合、すべての SQL Anywhere データベースでバックアップに VSS サービスを使用できます。VSS は、SQL

Anywhere VSS ライターなしでデータベースのバックアップに使用できます。ただし、これらのデータベースをリストアするには、SQL Anywhere の完全なリカバリ手順を使用する必要がある可能性があります。データベースサーバーが VSS サービスに参加しないようにするには、データベースサーバーの起動時に **-vss-** を指定します。また、Windows 用サービスユーティリティ (dbsvc) を使用して VSS サービスの開始時刻を指定することもできます。

SQL Anywhere では VSS は次のように機能します。

- バックアップアプリケーションが、スナップショットを作成するコマンドを VSS に送信します。
- VSS が SQL Anywhere VSS ライター (*dbvss12.exe*) に 「identify」 コマンドを発行します。
- VSS が、すべてのトランザクションをサスペンドし、すべてのデータベースサーバーにあるすべてのデータベースの変更されたすべてのページをディスクに書き込む「prepare to snapshot」コマンドを発行します。データベースのトランザクションが 10 秒以内にサスペンドしなかった場合は、コミットされていないトランザクションがスナップショットに含まれ、フルリカバリが必要になる場合があります。
- VSS が、すべてのデータベースサーバーにあるすべてのデータベースのすべてのアクティビティをチェックポイントしてからサスペンドする 「freeze」 コマンドを発行します。各 SQL Anywhere データベースサーバーは、すべてのデータベースですべてのアクティビティがサスペンドされるまで最大 60 秒間待ちます。一般にこの処理には数秒間かかります。
- VSS が、SQL Anywhere VSS ライターに 「thaw」 コマンドを発行し、すべてのデータベースサーバーにあるすべてのデータベースに対するすべてのトランザクションを再開します。

まれに、VSS で許可されている最大時間内に、SQL Anywhere がトランザクションをサスペンドしたり、チェックポイントを完了したりできないことがあります。この場合は、トランザクションログファイルとフルリカバリ処理を使用して、バックアップしたデータベースをリカバリする必要があります。

**参照**


- 「Windows 用サービスユーティリティ (dbsvc)」1013 ページ
- 「Windows サービスの作成」172 ページ


# クライアント側バックアップ

クライアント側バックアップは、次の方法で作成できます。

| ツール | その他の情報 |
| --- | --- |
| バックアップユーティリティ (dbbackup) | バックアップユーティリティ (dbbackup) は、物理デバイスレベルの並列処理を使用して、バックアップ操作の完了に必要な総時間を節約します。並列バックアップは、Windows Mobile ではサポートされていません。 |
| DBBackup 関数 | DBBackup メソッド [データベースツール]『SQL Anywhere サーバー プログラミング』を参照してください。 |

**♦ クライアント側バックアップを作成するには、次の手順に従います (dbbackup ユーティリティの場合)。**

dbbackup ユーティリティでは、データベースファイルとトランザクションログ (任意) のコピーをそれぞれ別のファイルとして含むイメージバックアップが作成されます。

1. BACKUP 権限または REMOTE DBA 権限を持つユーザーでデータベースに接続します。

2. クライアントコンピューターでバックアップユーティリティ (dbbackup) を実行します。次に例を示します。

```
dbbackup -c "HOST=myhost;DBN=demo;UID=DBA;PWD=sql" SQLAnybackup
```

**参照**

- 「並列データベースバックアップ」1116 ページ
- 「バックアップユーティリティ (dbbackup)」913 ページ
- 「サーバー側のバックアップ」1079 ページ
- 「データベースのリカバリ」1089 ページ
- 「バックアップの種類」1072 ページ

## ライブバックアップの作成

ライブバックアップを使用すると、トランザクションログの冗長コピーを作成できます。作成したコピーは、データベースサーバーを実行している第 1 システムが使用できなくなった場合に、第 2 システムの再起動に使用されます。ライブバックアップは継続的に実行され、サーバーが停止した場合にのみ中止されます。システム障害が発生した場合は、バックアップされたトランザクションログを使って、システムをすばやく再起動できます。しかし、サーバーが処理するロード量によってライブバックアップが遅れ、コミットされたすべてのトランザクションがバックアップされないことがあります。

バックアップ (dbbackup) ユーティリティはセカンダリコンピューターから実行してください。プライマリコンピューターを使用できなくなった場合、セカンダリコンピューターを使用してデータベースを再起動できます。データベースファイルとトランザクションログは、再起動するのに必要な情報を保持しています。

dbbackup ユーティリティに -l オプションを指定すると、トランザクションログのライブバックアップを実行できます。

**♦ ライブバックアップを作成するには、次の手順に従います (dbbackup ユーティリティの場合)。**

1. オンラインコンピューターで障害が発生したときにデータベースを実行できるセカンダリコンピューターを設定します。SQL Anywhere がセカンダリコンピューター上にインストールされていることを確認します。

2. BACKUP 権限または REMOTE DBA 権限を持つユーザーでデータベースに接続します。

3. セカンダリコンピューターから、セカンダリコンピューターにフルバックアップを実行します。

   例：

   ```
   dbbackup -c "UID=DBA;PWD=sql;HOST=myhost;Server=testsrv;DBN=test" c:¥path-on-secondary-computer¥backup
   ```

4. セカンダリコンピューターから、セカンダリコンピューターにトランザクションログのライブバックアップを実行します。

   ```
   dbbackup -l path-on-secondary-computer¥filename.log -c "connection-string"
   ```

5. セカンダリコンピューターから、定期的にオンラインデータベースのフルバックアップを実行します。

   プライマリコンピューターを使用できなくなった場合、セカンダリコンピューターを使用してデータベースを再起動できます。データベースファイルとトランザクションログは、再起動するのに必要な情報を保持しています。

**参照**


- 「ライブバックアップ」1075 ページ
- 「データベースのリカバリ」1089 ページ
- 「ライブバックアップからの再起動」1093 ページ
- 「バックアップの種類」1072 ページ


# バックアップの検証

データベースファイルの破損は、データベースサーバーがデータベース内の破損部分にアクセスするまで判明しないことがあります。バックアップとリカバリのプランの一環として、Sybase Central の**データベース検証ウィザード**や、検証ユーティリティ (dbvalid) などのツールを使用して、データベースが有効であることを定期的に確認するようにしてください。データベースの検証は、バックアップの前後両方に実行してください。検証を実行するには、VALIDATE 権限が必要です。 「VALIDATE 権限」541 ページを参照してください。

検証のためにデータベースのバックアップコピーを起動するときは、-ds データベースオプションを使用して、DB 領域ファイルとトランザクションログのロケーションを指定できます。そう

することで、元のデータベースを稼働させたままで、データベースのバックアップコピーを元のデータベースと同じコンピューター上で起動できます。「-ds dbeng12/dbsrv12 データベースオプション」308 ページを参照してください。

オプションの指定によって、チェックサム、インデックスデータの正当性、すべてのテーブルページがデータベース内のオブジェクトに属するかどうかを検証できます。データベースのエクスプレス検証 (-fx オプション) では、データ、連続したローの構造、外部キー関係は検証されません。

**警告**
データベースとトランザクションログのバックアップコピーには、どのような変更でも加えるべきではありません。バックアップ中に処理中のトランザクションがなかった場合、または BACKUP DATABASE WITH CHECKPOINT LOG RECOVER か WITH CHECKPOINT LOG NO COPY を指定した場合は、読み込み専用モードを使用するか、バックアップデータベースのコピーを検証して、バックアップデータベースの妥当性をチェックできます。

一方、トランザクションの処理中だった場合、または BACKUP DATABASE WITH CHECKPOINT LOG COPY を指定した場合は、検証の開始時にデータベースサーバーがデータベースのリカバリを実行する必要が生じます。リカバリを実行するとバックアップコピーに変更が加えられますが、これは望ましいことではありません。

バックアップが作成されているときに実行中のトランザクションがなかったという確信が持てる場合は、データベースサーバーによるリカバリの手順を実行する必要はありません。代わりに、読み込み専用のデータベースオプションを使用して、バックアップしたデータベースに妥当性検査を実行できます。「-r dbeng12/dbsrv12 サーバーオプション」267 ページを参照してください。

**ヒント**
WAIT BEFORE START 句を使用して BACKUP 文を実行すると、トランザクションの処理中にはバックアップが開始されません。

検証を実行するには、検証するオブジェクトに対する排他的なアクセスが必要です。このことを考慮すれば、データベース上に他のアクティビティがない場合に、データベースの検証を行うのが最適です。

データベースファイル内のベーステーブルが破損している場合は、メディア障害として対処し、前のバックアップからリカバリしてください。インデックスが破損している場合は、インデックスなしでデータベースをアンロードして、再ロードします。

**参照**


- 「データベースの検証」1118 ページ
- 「トランザクションログの検証」1112 ページ
- 「テーブルの検証」1122 ページ
- 「VALIDATE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「データベースのリカバリ」1089 ページ

# データベースのリカバリ

「リカバリ」は、インクリメンタルトランザクションログファイルを使用してデータベースファイル、トランザクションログ、DB 領域をリストアし、データベースファイルをできるだけ最新の状態にすることです。

バックアップは、バックアップとリカバリのプランの一環として検証することが重要です。リカバリには、必ずデータベースの有効なバックアップコピーを使用してください。

リカバリ処理で実行する必要がある手順は、バックアップ処理でインクリメンタルバックアップのトランザクションログに変更を加えなかったかどうかによって異なります。バックアップの処理中にトランザクションログの削除または名前の変更を行った場合、複数のトランザクションログに加えられた変更を適用する必要があります。バックアップ処理でトランザクションログに変更が加えられていない場合は、リカバリのときにはオンライントランザクションログだけを使用すれば済みます。

トランザクションログが複数ある場合、トランザクションが複数のトランザクションログにまたがっている可能性があります。リカバリのときには、トランザクションログを正しい順序で適用する必要があります。適用しない場合、複数のトランザクションログにまたがっているトランザクションがロールバックされます。トランザクションログの正しい適用順序をデータベースサーバーに判断させる場合は、データベースサーバーオプションとして -ad を指定します。

**参照**

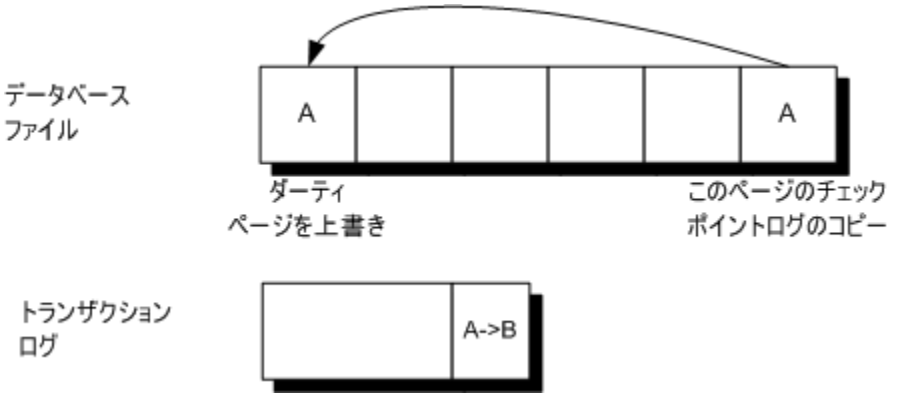
- 「トランザクションログが複数の場合のデータベースのリカバリ」1094 ページ

# 自動リカバリ処理

正常なオペレーションでデータベースが停止すると、データベースサーバーでチェックポイントが実行され、データベース内のすべての情報がデータベースファイル内に格納されます。これは、「クリーン」な停止と呼ばれます。

データベースを起動するたびに、データベースサーバーは最後の停止がクリーンだったのか、システム障害の結果だったのかをチェックします。データベースの停止がクリーンでなかった場合は、システム障害からリカバリするために、次の手順が自動的に実行されます。

1. **最新のチェックポイントにリカバリする**　すべてのページを最新のチェックポイント時の状態にリストアするために、チェックポイントログページのコピーでチェックポイント以降に加えられた変更が上書きされます。




2. **チェックポイント以降に加えられた変更を適用する**　チェックポイントからシステム障害が発生するまでの間に加えられた変更が適用されます。この変更内容はトランザクションログに格納されています。

3. **コミットされていないトランザクションのロールバック**　コミットされていないトランザクションが、ロールバックログを使用してロールバックされます。

## コミットされていない操作のリカバリ

データベースファイルのメディア障害からリカバリする場合、トランザクションログには影響ありません。リカバリを実行すると、すべてのコミットされたトランザクションがデータベースに再適用されます。状況によっては、障害が発生した時点で終了していなかったトランザクションについての情報を検索することも可能です。

**ログファイル変換ウィザード**を使用すると、Sybase Central でログファイルを *.sql* ファイルに変換できます。dbtran ユーティリティを使用して、ログファイルを *.sql* ファイルに変換することもできます。

**注意**
トランザクションログには、障害が発生する直前の変更が含まれていないこともあります。最後にコミットしたトランザクションより前にデータベースに加えられた変更は、トランザクションログに含まれます。

**♦ トランザクションログからコミットされていない操作をリカバリするには、次の手順に従います (Sybase Central の場合)。**

1. **[ツール]** » **[SQL Anywhere 12]** » **[ログファイルの変換]** をクリックします。

2. ウィザードの指示に従います。

3. 変換されたログ (SQL スクリプトファイル) をテキストエディターで編集し、必要な指示を特定します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**♦ トランザクションログからコミットされていない操作をリカバリするには、次の手順に従います (コマンドラインの場合)。**

1. dbtran を実行し、トランザクションログを SQL スクリプトファイルに変換します。このとき、-a オプションを指定して、コミットされていないトランザクションも含まれるようにします。たとえば、次のコマンドは、dbtran を使用してトランザクションログを変換します。

   ```
   dbtran -a sample.log changes.sql
   ```

2. 変換されたログ (SQL スクリプトファイル) をテキストエディターで編集し、必要な指示を特定します。

   ログ変換ユーティリティの詳細については、「ログ変換ユーティリティ (dbtran)」985 ページを参照してください。

**参照**


- 「イメージバックアップからのリストア」1091 ページ
- 「アーカイブバックアップからのリストア」1091 ページ
- 「ライブバックアップからの再起動」1093 ページ
- 「トランザクションログが複数の場合のデータベースのリカバリ」1094 ページ
- 「メディア障害リカバリ」1097 ページ


## イメージバックアップからのリストア

**♦ イメージバックアップからデータベースをリストアするには、次の手順に従います。**

この手順では、リカバリ処理中に適用する必要がある、トランザクションログのインクリメンタルバックアップがないと想定しています。トランザクションログの複数のコピーをバックアップした場合のデータベースのリカバリについては、「トランザクションログが複数の場合のデータベースのリカバリ」1094 ページを参照してください。

1. データベースファイルを元のロケーションにコピーします。

2. データベースサーバーを再起動します。

**参照**


- 「イメージバックアップ」1077 ページ
- 「自動リカバリ処理」1089 ページ
- 「コミットされていない操作のリカバリ」1090 ページ
- 「バックアップの種類」1072 ページ


## アーカイブバックアップからのリストア

**♦ アーカイブバックアップからデータベースをリストアするには、次の手順に従います (Sybase Central の場合)。**

この手順では、リカバリ処理中に適用する必要がある、トランザクションログのインクリメンタルバックアップがないと想定しています。トランザクションログの複数のコピーをバックアッ

プした場合のデータベースのリカバリについては、「トランザクションログが複数の場合のデータベースのリカバリ」1094 ページを参照してください。

空きページの除去をオンにして強力に暗号化されているデータベースのコピーを作成した場合は、データベースをリストアするときに暗号化キーを指定する必要があります。

1. パーソナルデータベースサーバーを起動します。

   たとえば、次のコマンドは、restore というデータベースサーバーを起動します。

   ```
   dbeng12 -n restore
   ```

2. Sybase Central を起動し、ユーティリティデータベースに接続します。次のフィールドを完成させます。
   - **[ユーザー ID]**　**DBA** と入力します。
   - **[パスワード]**　**sql** と入力します。
   - **[データベース名]**　**utility_db** と入力します。
3. **[OK]** をクリックします。
4. **[ツール]** » **[SQL Anywhere]12** » **[データベースのリストア]** をクリックします。
5. ウィザードの指示に従います。

**♦ アーカイブバックアップからデータベースをリストアするには、次の手順に従います (Interactive SQL の場合)。**

1. パーソナルデータベースサーバーを起動します。

   たとえば、次のコマンドは、restore というデータベースサーバーを起動します。

   ```
   dbeng12 -n restore
   ```

2. Interactive SQL を起動して、ユーティリティデータベースに接続します。次のフィールドを完成させます。
   - **[ユーザー ID]**　**DBA** と入力します。
   - **[パスワード]**　**sql** と入力します。
   - **[データベース名]**　**utility_db** と入力します。
3. **[OK]** をクリックします。
4. アーカイブルートを指定して RESTORE DATABASE 文を実行します。

   この時点で、アーカイブされたデータベースを元のロケーション (デフォルト) にリストアするか、RENAME 句を使用して別のコンピューターに別のデバイス名を使用してリストアするかを選択できます。「RESTORE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**参照**


- 「ユーティリティデータベース」135 ページ
- 「アーカイブバックアップ」1077 ページ
- 「自動リカバリ処理」1089 ページ
- 「コミットされていない操作のリカバリ」1090 ページ
- 「バックアップの種類」1072 ページ


**例**

次の文では、データベースをテープアーカイブからデータベースファイル *c:¥newdb¥newdb.db* にリストアします。

```
RESTORE DATABASE 'c:¥¥newdb¥¥newdb.db'
FROM '¥¥¥¥.¥¥tape0';
```

次の文では、データベースをファイル *c:¥backup¥archive.1* のアーカイブバックアップからデータベースファイル *c:¥newdb¥newdb.db* にリストアします。トランザクションログの名前と場所はデータベース内に指定されます。

```
RESTORE DATABASE 'c:¥¥newdb¥¥newdb.db'
FROM 'c:¥¥backup¥¥archive';
```

次の文では、データベースをファイル *c:¥backup¥archive.1* のアーカイブバックアップからデータベースファイル *c:¥newdb¥newdb.db* にリストアします。トランザクションログの名前と場所はデータベース内に指定されます。データベースの暗号化キーが指定されます。

```
RESTORE DATABASE 'c:¥¥newdb¥¥newdb.db'
FROM 'c:¥¥backup¥¥archive'
KEY '3Km57y1z';
```

## ライブバックアップからの再起動

**♦ ライブバックアップを使用してデータベースを再起動するには、次の手順に従います。**

ライブバックアップは、運用データベースを実行するプライマリコンピューターとは別のコンピューターに作成されます。ライブバックアップからデータベースを再起動するには、セカンダリコンピューターに SQL Anywhere をインストールする必要があります。「ライブバックアップ」1075 ページを参照してください。

1. フルバックアップトランザクションログファイルとライブバックアップトランザクションログを、データベースファイルのバックアップコピーに適用できるディレクトリにコピーします。

2. 現在のトランザクションログファイル名が予期されるトランザクションログファイル名と一致する場合は、名前を変更するか削除します。

3. データベースサーバーを -ad オプションを指定して起動し、手順 1 で作成したディレクトリにあるトランザクションログを適用し、データベースを最新にします。

   ```
   dbeng12 "%SQLANYSAMP12%¥demo.db" -ad directory-name
   ```

   データベースサーバーは、トランザクションログが適用されると自動的に停止します。

4. データベースサーバーを通常どおり起動して、ユーザーアクセスを許可します。新しいアクティビティは、すべて新しいトランザクションログに書き込まれます。

5. セカンダリコンピューターにトランザクションログのライブバックアップを実行します。

```
dbbackup -l path¥filename.log -c "connection-string"
```

**参照**


- 「ライブバックアップ」1075 ページ
- 「自動リカバリ処理」1089 ページ
- 「コミットされていない操作のリカバリ」1090 ページ
- 「バックアップの種類」1072 ページ


## トランザクションログが複数の場合のデータベースのリカバリ

データベースをリカバリする必要があり、複数のトランザクションログがある場合は、正しい順序でトランザクションログファイルをバックアップコピーに適用します。

トランザクションログを正しい順序で適用するには、次のいずれかの方法を使用します。

- -a サーバーオプションを使用して、各ログをデータベースのバックアップコピーに個別に適用します。トランザクションログユーティリティ (dblog) を使用して、トランザクションログファイルを生成する順番を判断します。このユーティリティは、トランザクションログの開始時のログオフセットが表示されます。これによって複数のログファイルを適用する順序を効果的に判断できます。

- -ad サーバーオプションを使用して、トランザクションログファイルのロケーションを指定します。データベースサーバーは、データベースのバックアップコピーに対するトランザクションログの正しい適用順序をログオフセットに基づいて判断します。

- -ar サーバーオプションを使用して、トランザクションログと同じディレクトリにある、データベースに関連付けられているログファイルを適用することを、データベースサーバーに指示します。トランザクションログのディレクトリはデータベースから取得されます。データベースサーバーは、データベースのバックアップコピーに対するトランザクションログの正しい適用順序をログオフセットに基づいて判断します。

- ログ変換ユーティリティ (dbtran) を使用して、1 つ以上のトランザクションログを、データベースのバックアップコピーに適用可能な *.sql* ファイルに変換します。

**参照**


- 「-a dbeng12/dbsrv12 データベースオプション」305 ページ
- 「-ad dbeng12/dbsrv12 データベースオプション」306 ページ
- 「-ar dbeng12/dbsrv12 データベースオプション」307 ページ
- 「トランザクションログユーティリティ (dblog)」1041 ページ


 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## -ad サーバーオプションを使用した、複数のトランザクションログがあるデータベースのリカバリ

-ad サーバーオプションを使用して、指定したディレクトリにあるすべてのトランザクションログファイルをデータベースのバックアップコピーに適用することで、データベースをリカバリします。このオプションを指定すると、データベースサーバーはトランザクションログを適用した後、データベースを停止します。

### 例

次の例では、-ad データベースサーバーオプションを使用して、オフライン (バックアップ) および現在のトランザクションログをサンプルデータベースのバックアップコピーに適用します。データベースサーバーは、トランザクションログのログオフセットを使用して、ログファイルの正しい適用順序を判断します。

1. バックアップトランザクションログと現在のトランザクションログをディレクトリにコピーします。たとえば、ファイルを *c:¥backuplogs* にコピーします。

2. データベースサーバーを起動し、トランザクションログを *backupdemo.db* というデータベースのバックアップコピーに適用します。

   ```
   dbeng12 backupdemo.db -ad c:¥backuplogs
   ```

   データベースサーバーは、トランザクションログをデータベースのバックアップコピーに適用した後、停止します。

### 参照


- 「-ad dbeng12/dbsrv12 データベースオプション」306 ページ
- 「-a サーバーオプションを使用した、複数のトランザクションログがあるデータベースのリカバリ」1095 ページ
- 「dbtran ユーティリティを使用した、複数のトランザクションログがあるデータベースのリカバリ」1096 ページ
- 「自動リカバリ処理」1089 ページ
- 「コミットされていない操作のリカバリ」1090 ページ


## -a サーバーオプションを使用した、複数のトランザクションログがあるデータベースのリカバリ

-a サーバーオプションを使用して単一のトランザクションログファイルをデータベースのバックアップコピーに適用することで、データベースをリカバリします。このオプションを指定すると、データベースサーバーはログを適用した後、停止します。複数のトランザクションログがある場合、ファイルを 1 つずつ正しい順序 (古いものから最新のものへ) で適用する必要があります。

### 例

次の例では、-a データベースサーバーオプションを使用して、オフライン (バックアップ) および現在のトランザクションログをサンプルデータベースのバックアップコピーに適用します。

1. データベースサーバーを起動し、*backupdemo.log* というバックアップトランザクションログを *backupdemo.db* というデータベースのバックアップコピーに適用します。

   ```
   dbeng12 backupdemo.db -a backupdemo.log
   ```

   データベースサーバーがバックアップトランザクションログをデータベースのバックアップコピーに適用した後、停止します。

2. データベースサーバーを起動して、*demo.log* という現在のトランザクションログをデータベースのバックアップコピーに適用します。

   ```
   dbeng12 backupdemo.db -a demo.log
   ```

   データベースサーバーが現在のトランザクションログをデータベースのバックアップコピーに適用した後、停止します。

**参照**


- 「-a dbeng12/dbsrv12 データベースオプション」305 ページ
- 「-ad サーバーオプションを使用した、複数のトランザクションログがあるデータベースのリカバリ」1095 ページ
- 「dbtran ユーティリティを使用した、複数のトランザクションログがあるデータベースのリカバリ」1096 ページ
- 「自動リカバリ処理」1089 ページ
- 「コミットされていない操作のリカバリ」1090 ページ


## dbtran ユーティリティを使用した、複数のトランザクションログがあるデータベースのリカバリ

**♦ 複数のトランザクションログからリカバリするには、次の手順に従います (dbtran ユーティリティ)。**

> **注意**
> ログ変換ユーティリティ (dbtran) を使用して複数のトランザクションログがあるデータベースのリカバリを行うよりも、-a または -ad を使用してログを適用することをおすすめします。
>
> ログ変換ユーティリティ (dbtran) は、最後のオプションとして使用するべきです。

dbtran により複数のトランザクションログを変換し、データの整合性を維持するには、-m および -n オプションの両方を指定する必要があります。-m オプションは、指定したディレクトリ内にあるログからのすべてのトランザクションを含むファイル (-n によって指定) を作成するようログ変換ユーティリティ (dbtran) に指示します。

複数のトランザクションログファイルにまたがるトランザクションがある場合、dbtran を使用して各ログを個別に変換すると、トランザクションがロールバックされる可能性があるため、-m を使用する必要があります。dbtran は、ログを変換するときに、ROLLBACK 文をログの最後に追加してコミットされていないトランザクションを取り消そうとします。トランザクションが 2 つのログにまたがっている場合、トランザクションの COMMIT が 2 番目のログファイルで発生します。最初のログファイルにトランザクションの COMMIT が含まれていないために、このファイルに対する最後の操作が dbtran によってロールバックされます。-m を使用してディレク

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

トリ内のすべてのトランザクションログファイルを変換すると、確実にすべてのトランザクションが変換されます。

1. トランザクションログファイルを含むディレクトリに対してログ変換ユーティリティ (dbtran) を実行して、その結果生成された SQL 文を *.sql* ファイルに出力します。

2. データベースのバックアップコピーを開始します。

3. 手順 1 で dbtran によって生成された *.sql* ファイルを、Interactive SQL からデータベースのバックアップコピーに適用します。

**参照**


- 「トランザクションログユーティリティ (dblog)」1041 ページ
- 「-ad サーバーオプションを使用した、複数のトランザクションログがあるデータベースのリカバリ」1095 ページ
- 「-a サーバーオプションを使用した、複数のトランザクションログがあるデータベースのリカバリ」1095 ページ
- 「自動リカバリ処理」1089 ページ
- 「コミットされていない操作のリカバリ」1090 ページ


**例**

次の例では、dbtran ユーティリティを使用してバックアップおよび現在のトランザクションログをデータベースのバックアップコピーに適用します。

1. *c:¥backup* ディレクトリに対してログ変換ユーティリティを実行して、SQL 文を *recoverylog.sql* というファイルに出力します。

```
dbtran -m "c:¥backup" -n recoverylog.sql
```

2. *backupdemo.db* というデータベースのバックアップコピーを起動します。

```
dbeng12 backupdemo.db
```

3. Interactive SQL から *recoverylog.sql* ファイルをデータベースに適用します。

```
dbisql -c "UID=DBA;PWD=sql;Server=backupdemo" READ recoverylog.sql
```

## メディア障害リカバリ

データベースが使用できなくなった場合、データベースの「障害」が発生しています。SQL Anywhere では、次の種類の障害に対する防護策を備えています。

| 障害の種類 | 説明 | 例 |
| --- | --- | --- |
| メディア | データベースファイルまたはトランザクションログが使用できない状態を指します。この種類の障害は、データベースファイルを格納しているファイルシステムまたはデバイスが使用できなくなった場合や、ファイルが破損した場合などに発生します。バックアップによって、メディア障害からデータを保護できます。 | ●データベースファイルまたはトランザクションログファイルが保存されているディスクドライブが使用できなくなった。<br><br>●データベースファイルまたはトランザクションログファイルが破損した。これはハードウェアまたはソフトウェアの問題によって発生することがあります。 |
| システム | システム障害は、トランザクションの途中で、コンピューターまたはオペレーティングシステムが停止した場合に発生します。この種類の障害は、正しい手順でコンピューターを終了または再起動しなかったり、他のアプリケーションによってオペレーティングシステムが停止したり、停電した場合に起こる可能性があります。<br><br>システム障害が発生した後、次にデータベースを起動するときに、データベースサーバーは自動的にリカバリします。システムエラー以前にコミットされたトランザクションの結果は保存されます。システム障害の前にコミットされていなかったトランザクションによる変更はすべてキャンセルされます。 | ●トランザクションが完了していないときに、停電やオペレーティングシステムの停止、またはコンピューターの不適切な再起動が原因で、コンピューターまたはオペレーティングシステムが一時的に使用できなくなった。 |

## データのメディア障害からのリカバリ

**♦ データベースファイルのメディア障害からリカバリするには、次の手順に従います。**

ここでは、失ったファイルがデータベースだけである場合にメディア障害からリカバリする手順について説明します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

1. 現在のトランザクションログの追加バックアップコピーを作成します。データベースファイルが使用できないので、最後のバックアップ後に行われた変更の記録は、唯一トランザクションログに含まれています。

2. リカバリの処理中に使用するファイルを保存する「リカバリディレクトリ」を作成します。

3. 最後のフルバックアップのデータベースファイルをリカバリディレクトリにコピーします。

4. バックアップされたトランザクションログに保持されているトランザクションをリカバリデータベースに適用します。次のいずれかの方法を使用します。

   各トランザクションログをログファイルごとに日付順に手動で適用するには、次の手順に従います。

   a. ログファイルをリカバリディレクトリにコピーします。

   b. 次のように、データベースサーバーをトランザクションログ適用 (-a) オプションを使用して起動し、トランザクションログを適用します。

      ```
      dbeng12 database-name.db -a log-name.log
      ```

      データベースサーバーは、トランザクションログが適用されると自動的に停止します。

   c. トランザクションログのバックアップをすべて適用したら、オンライントランザクションログをリカバリディレクトリにコピーします。

      オンライントランザクションログのトランザクションをリカバリデータベースに適用します。

      ```
      dbeng12 database-name.db -a log-name.log
      ```

   データベースサーバーでトランザクションログの正しい順序を判断して自動的に適用させるには、次の手順に従います。

   a. オンラインおよびオフラインのトランザクションログファイルをリカバリディレクトリにコピーします。

   b. データベースサーバーを -ad オプションを使用して起動し、トランザクションログのロケーションを指定します。データベースサーバーは、トランザクションログの正しい適用順序をログオフセットに基づいて判断します。

      ```
      dbeng12 database-name.db -ad log-directory
      ```

      データベースサーバーは、トランザクションログが適用されると自動的に停止します。

5. リカバリデータベースに対して妥当性検査を実行します。

6. バックアップを作成します。

7. データベースファイルを運用ディレクトリに移します。

8. 運用データベースにアクセスできることをユーザーに通知します。

**参照**


- 「データベースの検証」1118 ページ
- 「自動リカバリ処理」1089 ページ
- 「コミットされていない操作のリカバリ」1090 ページ


## トランザクションログミラーのメディア障害からのリカバリ

**♦ トランザクションログミラーのメディア障害からのリカバリ**

この手順では、トランザクションログミラーを使用している場合にメディア障害からリカバリする方法について説明します。データベースが SQL Remote インストール環境の統合データベースである場合、トランザクションログミラーまたは同様の機能を持つハードウェアを使用してください。

1. トランザクションログが開始された時点におけるデータベースファイルの追加バックアップコピーを作成します。

2. どちらのファイルが破損しているのか特定します。トランザクションログとそのミラーに対してログ変換ユーティリティ (dbtran) を実行します。エラーメッセージが生成されるファイルが破損しています。ログ変換ユーティリティには Sybase Central または dbtran ユーティリティからアクセスできます。

   次に示すコマンドラインは、トランザクションログ *demo.log* を変換して、*demo.sql* ファイルに出力します。

   ```
   dbtran demo.log
   ```

   ログ変換ユーティリティでは、正常なファイルは正しく変換し、破損したファイルを変換する際はエラーをレポートします。

3. 正しいファイルをコピーして、破損しているファイルに上書きします。

4. データベースサーバーを再起動します。

**参照**


- 「トランザクションログミラー」128 ページ
- 「自動リカバリ処理」1089 ページ
- 「コミットされていない操作のリカバリ」1090 ページ


## ミラーされていないトランザクションログのメディア障害からリカバリする

**♦ ミラーされていないトランザクションログのメディア障害からリカバリするには、次の手順に従います (部分リカバリの場合)。**

データベースが Mobile Link や SQL Remote インストール環境の統合データベースである場合、トランザクションログミラーまたは同様の機能を持つハードウェアを使用してください。

**警告**
dbeng12 コマンドは、データベースが Mobile Link や SQL Remote システムに関連していない場合にだけ使用してください。データベースが SQL Remote レプリケーションシステム内の統合データベースである場合は、リモートデータベースを再抽出する必要があります。

1. データベースファイルの追加バックアップコピーを作成します。トランザクションログがない場合、最後にバックアップが行われてから最新のチェックポイントまでの間に加えられた変更の記録は、唯一データベースファイルに含まれています。

2. トランザクションログファイルを削除するか、名前を変更します。

3. -f オプションを使って、データベースを再起動します。

   ```
   dbeng12 "%SQLANYSAMP12%¥demo.db" -f
   ```

   -f オプションを指定しないと、データベースサーバーはトランザクションログがないことを知らせるエラーメッセージを表示します。-f オプションを指定すると、データベースサーバーは最新のチェックポイント時の状態にデータベースをリストアし、チェックポイントの時点でコミットされていなかったトランザクションをすべてロールバックします。その後新しいトランザクションログが作成されます。

**参照**


- 「トランザクションログミラー」128 ページ
- 「トランザクションログ」127 ページ
- 「自動リカバリ処理」1089 ページ
- 「コミットされていない操作のリカバリ」1090 ページ
- 「-f dbeng12/dbsrv12 サーバーリカバリオプション」228 ページ


# バックアップとリカバリのプランの設計

データを保護するため、バックアップのスケジュールを立て、実行することをおすすめします。また、バックアップとリカバリのプランの一環として、バックアップとリカバリのコマンドを作成し、テストしてください。

バックアップとリカバリのプランを立てるときには、次のような点を検討する必要があります。

- データベースファイルの場所
- バックアップが必要なファイル
- バックアップファイルの保存場所
- データベースまたはアプリケーションのパフォーマンスに対するバックアップの影響
- データベースサーバーの稼働中にバックアップを実行するのか

一般に、バックアップは次の場合に必要です。

- メディア障害
- ハードウェア障害
- ファイル破損

一般的に、バックアップには、フルバックアップとインクリメンタルバックアップを組み合わせて使用します。各バックアップの頻度は、保護するデータの種類によって異なります。また、バックアップは、リカバリに使用できることを確認するために検証する必要があります。「バックアップの検証」1087 ページを参照してください。

SQL Anywhere のスケジュール機能を使用して、データベースをバックアップするタスクを自動化できます。スケジュールを指定しておくと、以後データベースサーバーによってバックアップが自動的に実行されます。「スケジュールとイベントの使用によるタスクの自動化」1123 ページと「メンテナンスプランの作成」1104 ページを参照してください。

データベースを使用する組織がどれだけの時間データベース内のデータにアクセスしなくても支障がないかによって、リカバリに割り当てられる時間の上限が決まります。

データベースファイルとトランザクションログファイルのメディア障害に対して、必要な保護措置が取られていることを確認してください。レプリケーション環境で作業している場合は、トランザクションログミラーの使用を検討してください。「メディア障害」1106 ページを参照してください。

リカバリの所要時間は、使用可能なハードウェア、データベースファイルのサイズ、リカバリに使用する媒体、ディスク領域、予期しないエラーなどの外部要因の影響を受けます。バックアップ方式を計画するときには、リカバリコマンドの入力、テープの検索やロードなどのタスクにかかる時間も考慮してリカバリの所要時間を決めてください。

**参照**


- 「バックアップコンポーネント」1113 ページ
- 「同期やレプリケーションに関連するデータベースのバックアップ」1107 ページ
- 「バックアップとリカバリの制限」1079 ページ
- 「バックアップの種類」1072 ページ
- 「バックアップフォーマット」1076 ページ
- 「フルバックアップ」1074 ページ
- 「インクリメンタルバックアップ」1074 ページ


## バックアップとリカバリ処理の設定

**♦ バックアップとリカバリのプランの実行**

1. データベース検証のコマンドを含め、バックアップとリカバリのコマンドを作成して検証します。

2. バックアップとリカバリのコマンド実行にかかる時間を測定します。

3. バックアップコマンドやバックアップの格納先を記述した手順書を作成します。手順書には、使用する命名規則、実行するバックアップの種類も明記しておきます。

4. 手順のとおりに運用サーバーにバックアップを設定します。

5. 予期しないエラーを防止するために、バックアップ手順をモニターします。手順を変更した場合は、必ず手順書にも反映するようにします。

**参照**


- 「バックアップの検証」1087 ページ
- 「フルバックアップ」1074 ページ
- 「インクリメンタルバックアップ」1074 ページ
- 「バックアップコンポーネント」1113 ページ
- 「同期やレプリケーションに関連するデータベースのバックアップ」1107 ページ
- 「バックアップとリカバリの制限」1079 ページ
- 「バックアップの種類」1072 ページ
- 「バックアップフォーマット」1076 ページ


## バックアップのスケジュール

一般に、バックアップには、フルバックアップとインクリメンタルバックアップを組み合わせて使用します。バックアップを作成する頻度は、データの重要性、データが変更される頻度などの要因によって異なります。

一般的には、週ごとにフルバックアップを行い、1 日 1 回トランザクションログのインクリメンタルバックアップを実行するというスケジュールから始めます。フルバックアップとインクリメンタルバックアップのどちらについても、オンライン (データベースの稼働中) またはオフラインで、サーバー側またはクライアント側で実行できます。

バックアップのスケジュールによってどの種類の障害からデータを保護できるかは、バックアップのスケジュールの頻度だけでなく、データベースサーバーの運用方法によっても異なります。

常に複数のフルバックアップを保管してください。また、火事、洪水、地震、盗難、その他の破壊行為に備えて、フルバックアップのコピーをオフサイトに保管してください。

SQL Anywhere のイベントスケジュール機能を使用して、スケジュールした時刻に自動的にオンラインバックアップを実行できます。

**警告**
以前のバックアップに上書きする形でバックアップを作成すると、バックアップの最中にメディア障害が発生した場合、バックアップは失われてしまいます。

**参照**


- 「メンテナンスプランの作成」1104 ページ

# メンテナンスプランの作成

管理を簡素化するには、データベースに対するメンテナンスプランを設定し、データベースサーバーで自動的に実行します。メンテナンスプランは、次のタスクを 1 つ以上実行するスケジュールで構成されます。

- データベースの検証
- データベースのバックアップ
- メンテナンスプランレポートの管理

メンテナンスプランは、Sybase Central で**メンテナンスプラン作成ウィザード**を使用して作成します。一度に実行できるメンテナンスプランのインスタンスは 1 つだけです。メンテナンスプランが実行されるたび、メンテナンスプランレポートがデータベースに保存されます。このレポートは Sybase Central で表示できます。また、データベースでメンテナンスプランが実行された後に、メンテナンスプランレポートを電子メールで受け取ることもできます。

「クラウドに関する注意点」この機能は、クラウドのテナントデータベースではサポートされていません。

### メンテナンスプランのカスタマイズ

メンテナンスプランには、ユーザー定義の操作を含めることができます。**メンテナンスプラン作成ウィザード**で、ユーザー定義の操作を、検証前またはバックアップ後に実行する SQL 文として追加できます。

### メンテナンスプランのステータスの概要

メンテナンスプランが実行中の場合は、Sybase Central の **[メンテナンスプラン]** フォルダーにステータスが表示されます。実行中のメンテナンスプランは、次のいずれかのステータスになります。

- **[初期化しています]** 新しい接続を無効にし、現在のユーザーを無効にします。
- **[事前検証を実行しています]** ユーザー定義の操作を実行します (メンテナンスプランの作成時に指定された場合)。
- **[ページを検証しています]** データベースページの検証のステータスを示します。
- **[オブジェクトを検証しています]** テーブル、マテリアライズドビュー、インデックスの検証のステータスを示します。
- **[バックアップしています]** バックアップの進行状況を示します。
- **[ポストバックアップを実行しています]** ユーザー定義の操作を実行します (メンテナンスプランの作成時に指定された場合)。
- **[終了しています]** 新しい接続をもう一度有効にし、レポートを保存、破棄、電子メールで送信します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## メンテナンスプランレポートの作成

♦ **メンテナンスプランを作成するには、次の手順に従います (Sybase Central の場合)。**

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。

2. 左ウィンドウ枠で **[メンテナンスプラン]** を右クリックし、**[新規]** » **[メンテナンスプラン]** をクリックします。

3. ウィザードの指示に従います。

   使用できる設定については、次の項を参照してください。

   - 「スケジュール定義」1126 ページ
   - 「VALIDATE 文」『SQL Anywhere サーバー SQL リファレンス』
   - 「バックアップの種類」1072 ページ
   - 「xp_startsmtp システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』

### 参照

- 「アーカイブバックアップ」1077 ページ
- 「イメージバックアップ」1077 ページ
- 「インクリメンタルバックアップ」1074 ページ
- 「メンテナンスプランの作成」1104 ページ

## メンテナンスプランレポートの表示

♦ **メンテナンスプランレポートを表示するには、次の手順に従います (Sybase Central の場合)。**

メンテナンスプランの実行後、Sybase Central でレポートを表示できます。

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。

2. 左ウィンドウ枠で、**[メンテナンスプラン]** をダブルクリックします。

3. 目的のメンテナンスプランをダブルクリックします。

4. 右ウィンドウ枠で、レポートをダブルクリックします。

   **[メンテナンスプランのプロパティ]** ウィンドウが表示されます。**[詳細]** ウィンドウ枠に、メンテナンスプランのログが表示されます。

### 参照

- 「アーカイブバックアップ」1077 ページ
- 「イメージバックアップ」1077 ページ
- 「インクリメンタルバックアップ」1074 ページ
- 「メンテナンスプランの作成」1104 ページ

# メディア障害

バックアップによって、メディア障害からデータを保護できます。

データベースを作成するとき、トランザクションログのデフォルトのロケーションは、データベースファイルと同じデバイス上の同じディレクトリです。この方法ではメディア障害からデータを保護できないので、運用上、トランザクションログを別の場所に格納してください。

**データベースファイルで発生したメディア障害**　データベースファイルが使用できず、トランザクションログを使用できる場合、所定の手順どおりに正しくバックアップしているかぎり、データベースにコミットされた変更はすべてリカバリできます。データベースファイルのコピーを最後にバックアップしてからの情報は、すべてバックアップされたトランザクションログまたはオンラインのトランザクションログに格納されています。

**トランザクションログファイルで発生したメディア障害**　トランザクションログミラーを使用しないかぎり、データベースチェックポイントを最後に実行してからトランザクションログでメディア障害が発生するまでに入力された情報はリカバリできません。このため、SQL Remote 統合データベースなどのセットアップでは、トランザクションログミラーを使用することをおすすめします。これらのセットアップでは、トランザクションログを失うと、重要な情報の損失や、レプリケーションシステムの破損につながる可能性があります。

メディア障害からのリカバリに要する時間は、メディア障害の発生場所がデータベースファイルなのかトランザクションログファイルなのかによって異なります。

メディア障害から確実に保護するには、データベースファイルと異なるデバイスにトランザクションログを保存してください。複数のハードドライブがあるコンピューターの中には、実際には 1 つの物理ストレージドライブをいくつかの論理ドライブまたはパーティションに区切っただけのものがあります。メディア障害に確実に備えるには、最低 2 つの物理ストレージデバイスがあるコンピューターを使用します。

トランザクションログを別のデバイスに格納すると、ディスクのヘッドが、トランザクションログとメインデータベースファイルの間を移動しなくて済むことからも、パフォーマンスの改善が期待できます。

**警告**
ネットワークディレクトリにトランザクションログを格納しないでください。ネットワークを介してページに対する読み取りと書き込みを行うと、パフォーマンスの低下やファイルの破損を招く可能性があります。

**参照**


- 「SQL Anywhere データベースの作成」113 ページ
- 「トランザクションログの場所の変更」129 ページ


 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

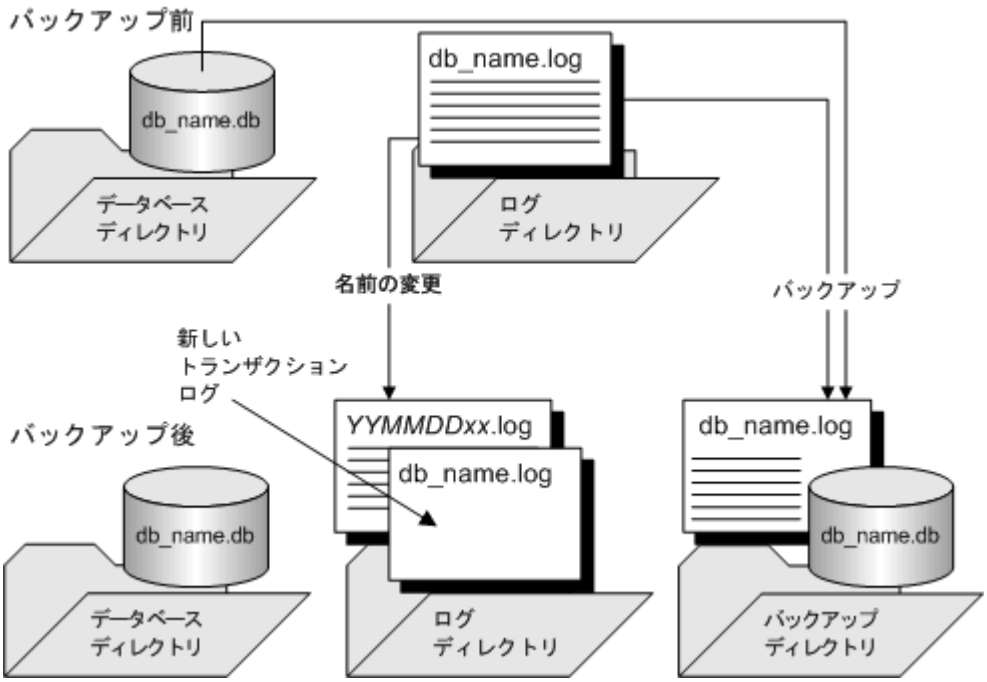
# 同期やレプリケーションに関連するデータベースのバックアップ

データベースが SQL Remote インストール環境の一部である場合、Message Agent は古いトランザクションにアクセスできる必要があります。統合データベースの場合、データベース内に SQL Remote インストール環境全体のマスターコピーが格納されるので、データを失わないためには完全なバックアップ手順が必要です。

データベースが Mobile Link の設定に関連していて、dbmlsync を使用している場合は、同じことに注意する必要があります。ただし、データベースが Mobile Link の統合データベースである場合は、古いトランザクションログは必要ありません。

同期やレプリケーションの環境では、トランザクションログの名前を変更してトランザクションログを再開するようにバックアップオプションを選択できます。このようなバックアップを実行すると、古いトランザクションの情報を保ちつつ、トランザクションログが無限に拡大するのを防ぐことができます。

この種類のバックアップを次の図で説明します。




詳細については、「バックアップを作成し、元のトランザクションログの名前を変更する」1108 ページを参照してください。

リモートデータベースでは、バックアップ手順は統合データベースの場合ほど重要ではありません。データのバックアップを、統合データベースへのレプリケーションに頼る方法もあります。ただし、メディア障害が発生した場合は、統合データベースからリモートデータベースを再抽出

しなければならず、レプリケートされていないオペレーションは失われますログ変換ユーティリティを使用して、失われたオペレーションのリカバリを実行することは可能です。「ログ変換ユーティリティ (dbtran)」985 ページを参照してください。

レプリケーションに頼ってリモートデータベースのデータを保護する場合でも、トランザクションログが大きくなりすぎるのを防ぐために、リモートデータベースでバックアップを定期的に実行する必要があります。統合データベースで使用するのと同じオプション (ログの名前変更と再起動) を使用して Message Agent を実行し、Message Agent が名前変更されたログファイルにアクセスできるようにします。リモートデータベースで delete_old_logs オプションを On に設定すると、不要になった古いログファイルが Message Agent によって自動的に削除されます。「delete_old_logs オプション [SQL Remote]」621 ページを参照してください。

### SQL Remote でのトランザクションログの自動名前変更

Message Agent の -x オプションを使用すると、データベースサーバーを停止した際に、リモートコンピューターでトランザクションログの名前を変更する必要がなくなります。-x オプションは、トランザクションログが出力メッセージ用にスキャンされた後で、トランザクションログの名前を変更します。「SQL Remote Message Agent ユーティリティ (dbremote)」『SQL Remote』を参照してください。

# トランザクションログの管理

## バックアップを作成し、元のトランザクションログの名前を変更する

データベースをバックアップするときは、引き続き既存のトランザクションログを使用するか、新しいトランザクションログを作成するかを決めます。データベースが同期またはレプリケーションに関連している場合は、必要がないことを確信できるまで、古いトランザクションログのコピーを維持してください。

**♦ バックアップを作成し、トランザクションログの名前を変更するには、次の手順に従います (Sybase Central の場合)。**

1. **SQL Anywhere 12** プラグインを使用して、バックアップ権限のあるユーザーとして、データベースに接続します。
2. データベースを右クリックし、**[バックアップイメージの作成]** をクリックします。
3. **[次へ]** をクリックします。
4. **[バックアップするデータベースを指定してください。]** リストでデータベースを選択し、**[次へ]** をクリックします。
5. **[バックアップイメージを次のディレクトリに保存]** フィールドに、バックアップコピーを保存するディレクトリの名前を入力します。
6. **[バックアップするファイルを指定してください。]** リストでオプションを選択し、**[次へ]** をクリックします。
7. **[トランザクションログの処理方法を指定してください。]** リストで、**[トランザクションログの名前を変更]** をクリックします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

8. **[次へ]** をクリックします。

9. **[完了]** をクリックします。

**♦ バックアップを作成し、トランザクションログの名前を変更するには、次の手順に従います (SQL の場合)。**

● 次の句を使用して BACKUP 文を実行します。

```
BACKUP DATABASE
DIRECTORY backup-directory
[ TRANSACTION LOG ONLY ]
TRANSACTION LOG RENAME;
```

インクリメンタルバックアップを作成する場合だけ TRANSACTION LOG ONLY 句を使用します。

トランザクションログとデータベースファイルのバックアップコピーは、*backup-directory* に格納されます。パスを入力する場合、クライアントアプリケーションではなく、データベースサーバーの作業フォルダーとの相対パスを入力します。

**♦ バックアップを作成し、トランザクションログの名前を変更するには、次の手順に従います (コマンドラインの場合)。**

● 次のコマンドを 1 行に入力して実行します。

```
dbbackup -c "connection-string" -r [ -t ] backup-directory
```

インクリメンタルバックアップを作成する場合は、-t オプションを使用します。

トランザクションログとデータベースファイルのバックアップコピーは、*backup-directory* に格納されます。パスを入力する場合、コマンドを実行するフォルダーとの相対パスを入力します。

**参照**


● 「バックアップユーティリティ (dbbackup)」 913 ページ
● 「BACKUP 文」『SQL Anywhere サーバー SQL リファレンス』
● 「トランザクションログ」 127 ページ
● 「データベースのリカバリ」 1089 ページ


## トランザクションログバックアップコピーに名前を付ける

デフォルトでは、トランザクションログファイルのバックアップコピーには、オンラインファイルと同じ名前が付けられます。バックアップを実行するたびに、バックアップコピーに別の名前または場所を割り当てるか、次のバックアップが終了する前にバックアップコピーを移動する必要があります。

繰り返し可能なインクリメンタルバックアップコマンドを作成するには、トランザクションログのバックアップコピーの名前を変更します。

**注意**
トランザクションログのバックアップコピーには、*YYMMDDxx.log* 形式の名前が付きます。*YY* は年、*MM* は月、*DD* は日です。*xx* は AA から ZZ までの英字で、1 日に何度もバックアップをする場合に値が 1 ずつ増加します。*YYMMDDxx.log* というファイル名は、順序付けではなく、ファイルを区別するために使用されます。

通常、この一連のバックアップオプションは、レプリケーションに関連するデータベースに使用します。データベースファイルのバックアップコピーとトランザクションログを作成するだけでなく、バックアップ時のトランザクションログはオフラインログとして、名前が変更されます。新しいトランザクションログは、バックアップ時と同じログ名になります。

**♦ トランザクションログのバックアップコピーの名前を変更するには、次の手順に従います (SQL の場合)。**

- BACKUP 文内に MATCH キーワードを使用します。たとえば、次の文では、トランザクションログのインクリメンタルバックアップをディレクトリ *c:¥backup* に作成します。トランザクションログのバックアップコピーには、*YYMMDDxx.log* 形式の名前が付きます。*YYMMDD* は日付、*xx* は AA から始まるカウンターです。

```
BACKUP DATABASE
DIRECTORY 'c:¥¥backup'
TRANSACTION LOG ONLY
TRANSACTION LOG RENAME MATCH;
```

**♦ トランザクションログのバックアップコピーの名前を変更するには、次の手順に従います (コマンドラインの場合)。**

- dbbackup に -n オプションを指定します。たとえば、次のコマンドでは、サンプルデータベースのインクリメンタルバックアップを作成し、トランザクションログのバックアップコピーの名前を変更します。

```
dbbackup -c "UID=DBA;PWD=sql;DBN=demo" -r -t -n c:¥backup
```

**参照**


- 「同期やレプリケーションに関連するデータベースのバックアップ」1107 ページ
- 「バックアップユーティリティ (dbbackup)」913 ページ
- 「BACKUP 文」『SQL Anywhere サーバー SQL リファレンス』
- 「バックアップを作成し、元のトランザクションログの名前を変更する」1108 ページ
- 「トランザクションログ」127 ページ
- 「データベースのリカバリ」1089 ページ


## バックアップを作成し、元のトランザクションログを削除する

データベースがレプリケーションに関連せず、コンピューター上のディスク領域に制限がある場合は、バックアップを作成するときにオンライントランザクションログの内容を削除 (ログを「トランケート」) できます。このようなバックアップを使用しているときにデータベースをリカバリするには、データベースファイルで発生したメディア障害からのリカバリ中に、最後のフルバックアップ以降に作成したすべてのバックアップコピーを使用する必要があります。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**♦ バックアップを作成し、トランザクションログを削除するには、次の手順に従います (Sybase Central の場合)。**

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。
2. データベースを右クリックし、**[バックアップイメージの作成]** をクリックします。
3. **[次へ]** をクリックします。
4. **[バックアップするデータベースを指定してください。]** リストでデータベースを選択し、**[次へ]** をクリックします。
5. **[バックアップイメージを次のディレクトリに保存]** フィールドに、バックアップコピーを保存するディレクトリの名前を入力します。
6. **[バックアップするファイルを指定してください。]** リストでオプションを選択し、**[次へ]** をクリックします。
7. **[トランザクションログの処理方法を指定してください。]** リストで、**[トランザクションログをトランケート]** をクリックします。
8. **[次へ]** をクリックします。
9. **[完了]** をクリックします。

**♦ バックアップを作成し、トランザクションログを削除するには、次の手順に従います (SQL の場合)。**

● 次の句を使用して BACKUP 文を実行します。

```
BACKUP DATABASE
DIRECTORY backup-directory
[ TRANSACTION LOG ONLY ]
TRANSACTION LOG TRUNCATE;
```

インクリメンタルバックアップを作成する場合だけ TRANSACTION LOG ONLY 句を使用します。

トランザクションログとデータベースファイルのバックアップコピーは、*backup-directory* に格納されます。パスを入力する場合、クライアントアプリケーションではなく、データベースサーバーの作業フォルダーとの相対パスを入力します。

**♦ バックアップを作成し、トランザクションログを削除するには、次の手順に従います (コマンドラインの場合)。**

● 次のコマンドを実行します。

```
dbbackup -c "connection-string" -x [ -t ] backup-directory
```

インクリメンタルバックアップを作成する場合だけ -t オプションを使用します。

トランザクションログとデータベースファイルのバックアップコピーは、*backup-directory* に格納されます。パスを入力する場合、コマンドを実行するフォルダーとの相対パスを入力します。

**参照**

- 「バックアップユーティリティ (dbbackup)」913 ページ
- 「BACKUP 文」『SQL Anywhere サーバー SQL リファレンス』
- 「トランザクションログ」127 ページ
- 「データベースのリカバリ」1089 ページ

## トランザクションログの検証

データベースサーバーは、トランザクションログミラーを使用するデータベースの起動時に一連の検証を行い、自動リカバリ操作を実行してトランザクションログとそのミラーが壊れていないかを確認します。壊れている場合は、いくつかの問題を修正します。

起動時には、トランザクションログとミラーを比較して、2 つのファイルが同一であるかを調べます。同一であれば、データベースは通常どおりに起動します。データベースの開始は、このログとミラーの比較のために時間がかかります。

システム障害のためにデータベースが停止した場合、操作の一部がトランザクションログには書き込まれても、ミラーには書き込まれていない可能性があります。トランザクションログとミラーのどちらか短い方のファイルを最後までチェックして 2 つのファイルが同じであることをサーバーが確認すると、長い方のファイルの残りの部分が短い方のファイルにコピーされます。これによって、ログとミラーは同一になります。このリカバリ作業の後に、サーバーは正常に起動します。

トランザクションログとトランザクションログミラーの内容が異なる場合は、どちらかのファイルが破損しています。この場合、データベースは起動しないで、エラーメッセージが表示され、トランザクションログかミラーのどちらかが無効であることを知らせます。

トランザクションログの検証には、対象となるトランザクションログがオンラインかオフラインかに関係なく、ログ変換ユーティリティ (dbtran) を使用することもできます。ログ変換ユーティリティによるログファイルの読み込みが正常に実行される場合、そのログは有効です。

**参照**

- 「ログ変換ユーティリティ (dbtran)」985 ページ

# バックアップの内部処理

バックアップを開始するとき、多くのユーザーがデータベースを使用中である可能性があります。バックアップからデータベースをリストアする必要がある場合は、どの情報がバックアップされて、どの情報がバックアップされていないかを把握しておく必要があります。

バックアップを作成すると、データベースサーバーで次の処理が実行されます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

1. チェックポイントを発行します。バックアップが完了するまで、これ以上チェックポイントは使用できません。

2. フルバックアップの場合は、データベースファイルのバックアップを作成します。

3. トランザクションログのバックアップを作成します。

   ログの最後のページが読み込まれる前に、トランザクションログに記録されたすべてのオペレーションがバックアップされます。これには、バックアップの開始後に発行された命令も含まれます。

   通常、トランザクションログのバックアップコピーは、オンライントランザクションログよりも小さくなります。データベースサーバーでは、オンライントランザクションログに 64 KB 単位で領域を割り当てるので、トランザクションログファイルのサイズには空のページも含まれるのが一般的です。しかし、空でないページだけがバックアップされます。

4. データベースのバックアップイメージにマークを付けて、リカバリが必要であることを示します。この処理によって、バックアップの開始以降に実行されたオペレーションが、データベースのバックアップコピーの開始時に適用されます。また、チェックポイントの時点で完了していなかったオペレーションは、コミットされていなければ取り消されます。

## バックアップコンポーネント

データベースが正常に停止された場合、データベースファイルにはデータベースにある全データの現行の完全なコピーが保持されています。しかし、データベースの稼働中は、データベースファイルは通常現行のものまたは完全なものではありません。

データベースファイルが全データの完全な現行のコピーを保持していることが保証されるのは、チェックポイントの完了直後だけです。チェックポイントの後は、データベースキャッシュの全内容がディスク上にあります。

データベースサーバーは、次の場合にデータベースに対するチェックポイントを実行します。

- データベースの停止操作の一環として
- 最後にチェックポイントからの経過時間が、-gc サーバーオプションの設定を超えたとき
- リカバリ予想時間が、-gr サーバーオプションの設定を超えたとき
- データベースサーバーのアイドル状態が、ダーティページをすべて書き込めるほどに長く続いているとき
- 特定の DDL 文 (ALTER TABLE、DROP TABLE、DROP INDEX、LOAD TABLE、BACKUP など) が実行されたとき
- 接続により CHECKPOINT 文が発行されたとき
- データベースサーバーがトランザクションログなしで稼働している場合に、トランザクションがコミットされたとき

チェックポイント間にコミットされたすべてのトランザクションの完全なコピーを維持するには、データベースファイルとトランザクションログの両方が必要です。

**参照**


- 「チェックポイントログ」133 ページ
- 「バックアップとリカバリの制限」1079 ページ
- 「データベースサーバーがチェックポイントのタイミングを決定する方法」1114 ページ
- 「-gc dbeng12/dbsrv12 サーバーオプション」232 ページ
- 「-gr dbeng12/dbsrv12 サーバーオプション」242 ページ


## データベースサーバーがチェックポイントのタイミングを決定する方法

最後のチェックポイント以降の時間と作業量の増加に伴い、ダーティページをディスクに書き込む優先度も増します。この優先度は、以下の要因によって決まります。

- **チェックポイントの緊急度**　最後のチェックポイント以降の経過時間を、データベースのチェックポイント時間の設定に対するパーセンテージで表したものです。チェックポイント間の最大時間は、-gc サーバーオプションまたは checkpoint_time データベースオプションを使って分単位で設定できます。-gc を指定した場合、データベース内の checkpoint_time オプションの設定は無視されます。

- **リカバリの緊急度**　データベースの障害が直ちに発生した場合のリカバリに必要な時間の推計です。システム障害が発生したときにリカバリにかける最大時間は、-gr サーバーオプションまたは recovery_time データベースオプションを使って分単位で設定できます。-gr を指定した場合、データベース内の recovery_time オプションの設定は無視されます。

チェックポイントとリカバリの緊急度の値は、データベースサーバーがダーティページの書き込みを行うだけのアイドル時間を持てない場合にのみ重要です。チェックポイントの間隔の下限値は、recovery_time と checkpoint_time オプションの組み合わせによって決まります。recovery_time オプションの設定に従うとチェックポイントの間隔が短すぎる場合、その設定は尊重されません。

チェックポイントの頻度が高いとシステム障害からのリカバリは速くなります。しかし、データベースエンジンがダーティページを書き出す作業が増えます。

データベースの他のアクティビティがあるためにダーティページ数が 0 になり、チェックポイントの緊急度が 33% 以上である場合、チェックポイントは自動的に発生します。

チェックポイントの緊急度とリカバリの緊急度の値は、チェックポイントが発生するまで増加を続け、チェックポイントが発生すると 0 に戻ります。

**参照**


- 「チェックポイントログ」133 ページ
- 「-gc dbeng12/dbsrv12 サーバーオプション」232 ページ
- 「checkpoint_time オプション」603 ページ
- 「-gr dbeng12/dbsrv12 サーバーオプション」242 ページ
- 「recovery_time オプション」669 ページ

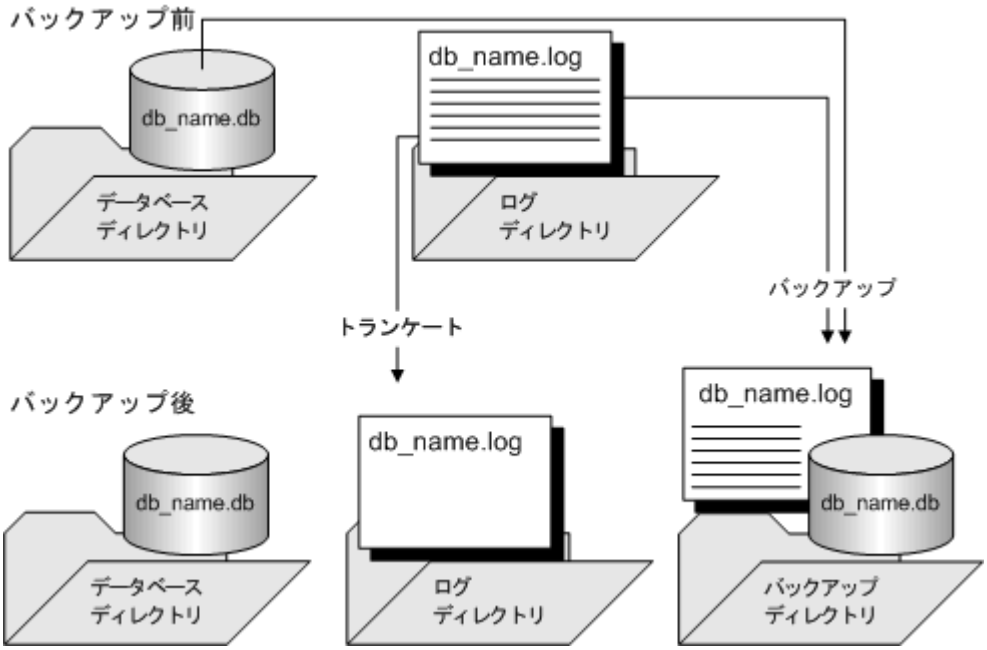

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

# トランザクションログの管理

バックアップを作成するとき、デフォルトでトランザクションログの現在の状態のコピーが作成され、トランザクションログはそのまま残されます。データベースが同期またはレプリケーションに関連している場合は、データベースのリカバリ後にトランザクションログの古いコピーにアクセスする必要がある場合があります。

多くの状況では、ディスク領域は制限されており、トランザクションログを無制限に増大させることは現実的ではありません。ディスク領域を解放するには、バックアップが完了したらトランザクションログの内容を削除することを選択できます。ただし、データベースがレプリケーションに関連する場合は、レプリケーションではトランザクションログへのアクセスを必要とするので、このオプションを選択しないでください。

次の図で、ログファイルをトランケートするフルバックアップを説明します。インクリメンタルバックアップでは、トランザクションログだけがバックアップされます。




各インクリメンタルバックアップ後にトランザクションログを削除すると、データベースファイルのメディア障害からリカバリするタスクが複雑になります。データベースを最新の状態にするには各トランザクションログを順番に適用する必要がありますが、最後のフルバックアップ以降に複数の異なるトランザクションログが作成されている可能性があります。

Mobile Link 統合データベースとして稼働するデータベースでは、このようなバックアップを実行できます。これは、Mobile Link サーバーがトランザクションログに依存しないためです。SQL Remote または Mobile Link *dbmlsync.exe* アプリケーションを実行している場合は、古いトランザクションログを保存するために、適切なスキームを使用してください。

**参照**


- 「バックアップを作成し、元のトランザクションログの名前を変更する」1108 ページ
- 「トランザクションログバックアップコピーに名前を付ける」1109 ページ
- 「バックアップを作成し、元のトランザクションログを削除する」1110 ページ


## オフライントランザクションログ

バックアップ操作では、トランザクションログのバックアップに加え、オンライントランザクションログのファイル名を *YYMMDDxx.log* という形式に変更できます。このファイルはデータベースサーバーでは使用されませんが、Message Agent では利用できます。これを「オフライン」トランザクションログと呼びます。新しいオンライントランザクションログの最初の名前には、古いオンライントランザクションログの名前が付けられます。

*YYMMDDxx.log* というファイル名は、順序付けではなく、ファイルを区別するために使用されます。たとえば、最初のバックアップが 2000 年 12 月 10 日である場合、ログファイル名は *001210AA.log* になります。最初の 2 桁は年、次の 2 桁は月、その次の 2 桁は日付を示し、最後の 2 文字によって、同じ日に実行された複数のバックアップを識別します。

Message Agent は、オフラインコピーを使用して古いトランザクションを必要に応じて提供できます。delete_old_logs データベースオプションを On に設定すると、Message Agent が必要としなくなった古いオフラインファイルは削除されるので、ディスク領域を節約できます。

## ロールバックログ

データベースの内容に変更が加えられると、「ロールバックログ」が作成されます。このログは、トランザクションがロールバックされた場合、またはシステム障害の発生時にトランザクションがコミットされていなかった場合に、変更をキャンセルするために使用されます。接続ごとに個別のロールバックログがあります。トランザクションのコミットまたはロールバックが行われると、その接続に関するロールバックログの内容は削除されます。ロールバックログは、データベースに格納されます。ロールバックログページは、変更される他のページとともにチェックポイントログにコピーされます。

ロールバックログは「取り消しログ」とも呼ばれます。

**参照**


- 「トランザクションと独立性レベル」『SQL Anywhere サーバー SQL の使用法』


# 並列データベースバックアップ

バックアップユーティリティ (dbbackup) を使用して -s オプションを指定するか、BACKUP DATABASE 文を使用して、サーバー側のイメージバックアップを実行すると、データベースの並列バックアップが実行されます。並列バックアップは、物理デバイスレベルの並列処理を使用して、バックアップ操作の完了に必要な総時間を節約します。並列バックアップは、Windows Mobile ではサポートされていません。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

データベースサーバーは、データベースファイルが保存されている各ドライブについてリーダースレッドを作成します。ライタースレッドは、バックアップディレクトリが存在するバックアップ先ドライブ用に作成されます。リーダーとライターを別々に使用することで、I/O 処理が順次的にではなく並列に実行されます。

並列バックアップのパフォーマンスは、システムで最も低速なコンポーネントによって制限されます。これは通常は物理ディスクですが、I/O コントローラやシステムバスなど、他のコンポーネントの場合もあります。これらの各コンポーネントのデータ転送レートには上限があります。

BACKUP DATABASE 文とバックアップユーティリティ (dbbackup) には、並列バックアップの動作を設定できるよう次のオプションが用意されています。

- チェックポイントログをコピーするタイミングと方法
- データベースサーバーから dbbackup へのデータ転送時に使用できる最大ページ数 (dbbackup の使用時のみ使用可)
- ライターの追加 (BACKUP 文の場合のみ)

バックアップは、必ず別の物理ドライブに作成してください。これにより、I/O 並列処理によるパフォーマンス改善が実現され、ハードウェア障害が発生したときのデータの安全性が向上します。

**参照**

- 「BACKUP 文」『SQL Anywhere サーバー SQL リファレンス』
- 「バックアップユーティリティ (dbbackup)」913 ページ

# データベースの検証

データベースファイルの破損は、データベースサーバーがデータベース内の破損部分にアクセスするまで判明しないことがあります。Sybase Central の**データベース検証ウィザード**や、検証ユーティリティ (dbvalid)、sa_validate システムプロシージャー、VALIDATE 文などのツールを使用して、データベースが有効であることを定期的に確認するようにしてください。検証を実行するには、VALIDATE 権限が必要です。「VALIDATE 権限」541 ページを参照してください。

オプションの指定によって、チェックサム、インデックスデータの正当性、すべてのテーブルページがデータベース内のオブジェクトに属するかどうかを検証できます。データベースのエクスプレス検証 (-fx オプション) では、データ、連続したローの構造、外部キー関係は検証されません。また、VALIDATE 文に WITH EXPRESS CHECK 句を指定することでも、参照整合性チェックを無効にできます。

検証は、検証対象のテーブルに対する排他アクセスを取得せず、検証中にテーブルが変更された場合には、破損をレポートします。このことを考慮すれば、データベース上に他のアクティビティがない場合に、データベースの検証を行うのが最適です。データベースファイル内のベーステーブルが破損している場合は、メディア障害として対処し、前のバックアップからリカバリしてください。インデックスが破損している場合は、インデックスなしでデータベースをアンロードしてから、再ロードします。また、ALTER INDEX 文の REBUILD 句を使用してインデックス

破損を修正することもできます。「ALTER INDEX 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

**ヒント**
WAIT BEFORE START 句を使用して BACKUP 文を実行すると、トランザクションの処理中にはバックアップが開始されません。バックアップ中に処理中のトランザクションがなかった場合、または BACKUP DATABASE WITH CHECKPOINT LOG RECOVER か WITH CHECKPOINT LOG NO COPY を指定した場合は、読み込み専用モードを使用するか、バックアップデータベースのコピーを検証して、バックアップデータベースの妥当性をチェックできます。「-r dbeng12/dbsrv12 サーバーオプション」267 ページを参照してください。

**参照**

- 「データベースの検証」1118 ページ
- 「トランザクションログの検証」1112 ページ
- 「テーブルの検証」1122 ページ
- 「VALIDATE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「sa_validate システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「検証ユーティリティ (dbvalid)」1064 ページ
- 「テキストインデックスの概念と参照」『SQL Anywhere サーバー SQL の使用法』

# データベースの検証

**注意**
バックアップコピーの妥当性を検証する場合は、どんな方法でも変更できないように、読み込み専用モードでデータベースを実行してください。バックアップ中に処理中のトランザクションがなかった場合にだけ、読み込み専用モードでデータベースを実行できます。「-r dbeng12/dbsrv12 サーバーオプション」267 ページを参照してください。

**♦ データベース全体の妥当性を検証するには、次の手順に従います (Sybase Central の場合)。**

**警告**
テーブルまたはデータベース全体の検証は、データベースに変更を加えている接続がない場合に実行してください。そうしないと、実際に破損していなくても、何らかの形でデータベースが破損したことを示すエラーがレポートされます。

1. **SQL Anywhere 12** プラグインを使用して、DBA または VALID 権限のあるユーザーとして、データベースに接続します。
2. 左ウィンドウ枠でデータベースを選択します。
3. **[ファイル]** メニューで、**[データベースの検証]** をクリックします。
4. **データベース検証ウィザード**の指示に従います。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**ヒント**
Sybase Central では、次の方法で**データベース検証ウィザード**を利用することもできます。

●データベースを右クリックし、**[データベースの検証]** をクリックする。

●データベースを選択し、**[ツール]** » **[SQL Anywhere 12]** » **[データベースの検証]** をクリックする。

**♦ データベース全体の妥当性を検証するには、次の手順に従います (SQL の場合)。**

**警告**
テーブルまたはデータベース全体の検証は、データベースに変更を加えている接続がない場合に実行してください。そうしないと、実際に破損していなくても、何らかの形でデータベースが破損したことを示すエラーがレポートされます。

● DBA 権限または VALID 権限を所有するユーザーとしてデータベースに接続します。

次のように sa_validate ストアドプロシージャーを実行します。

```
CALL sa_validate;
```

プロシージャーは Messages という 1 つのカラムを返します。すべてのテーブルが有効である場合、カラムには「**エラーは見つかりませんでした。**」と表示されます。

詳細については、「sa_validate システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

**♦ データベース全体の妥当性を検証するには、次の手順に従います (コマンドラインの場合)。**

**警告**
テーブルまたはデータベース全体の検証は、データベースに変更を加えている接続がない場合に実行してください。そうしないと、実際に破損していなくても、何らかの形でデータベースが破損したことを示すエラーがレポートされます。

● dbvalid ユーティリティを実行します。

```
dbvalid -c "connection-string"
```

DBA 権限または VALID 権限を所有するユーザーとしてデータベースに接続します。

**♦ データベース全体の妥当性を検証するには、次の手順に従います (Interactive SQL の場合)。**

**警告**
テーブルまたはデータベース全体の検証は、データベースに変更を加えている接続がない場合に実行してください。そうしないと、実際に破損していなくても、何らかの形でデータベースが破損したことを示すエラーがレポートされます。

1. DBA 権限または VALID 権限を所有するユーザーとしてデータベースに接続します。

2. VALIDATE DATABASE 文を実行します。

**参照**


- 「検証ユーティリティ (dbvalid)」1064 ページ
- 「VALIDATE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「VALIDATE 権限」541 ページ


# チェックサムを使用した破損の検出

「チェックサム」は、データベースページがディスク上で変更されたかどうかを判断するために使用します。SQL Anywhere では、グローバルチェックサムと書き込みチェックサムの 2 種類のチェックサムをサポートします。

## グローバルチェックサム

グローバルチェックサムは、データベースの作成時に有効にします。グローバルチェックサムを有効にしてデータベースを作成した場合、チェックサムはページがディスクに書き込まれる直前に計算されます。そのページが次にディスクから読み出されるときに、ページのチェックサムが再計算されて、ページに保存されているチェックサムと比較されます。チェックサムが異なる場合は、ディスク上でページが変更されており、エラーが発生します。

次の文を実行することによって、データベースがグローバルチェックサムを有効にして作成されたかどうかをチェックできます。

```
SELECT DB_PROPERTY ( 'Checksum' );
```

グローバルチェックサムが有効な場合、このクエリは ON を返します。それ以外の場合は OFF を返します。

ALTER DATABASE 文の CHECKSUM 句を使用すると、既存のデータベースのグローバルチェックサムを無効にすることができます。チェックサムを無効にすることはおすすめしません。グローバルチェックサムを一度無効にすると、有効にすることができなくなります。-wc+ オプションを使用してデータベースの書き込みチェックサムを有効にするか、またはチェックサムを有効にしてデータベースを再構築することができます。

## 書き込みチェックサム

デフォルトでは、バージョン 10 または 11 の SQL Anywhere で作成されたデータベースではグローバルチェックサムは有効になっていません。バージョン 12 のデータベースサーバーで SQL Anywhere 11 を使用して作成されたデータベースを起動すると、デフォルトでは、データベースサーバーで書き込みチェックサムが作成されます。書き込みチェックサムは、ページがディスクに書き込まれるときにのみページに追加されるチェックサムです。

バージョン 12 以降のデータベースでは、デフォルトでグローバルチェックサムが有効になっています。新しいデータベースを作成し、そのデータベースのグローバルチェックサムを無効にした場合は、-wc+ オプションまたは START DATABASE 文を使用して書き込みチェックサムを有効にすることができます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

次の文を実行することによって、データベースが書き込みチェックサムを有効にして起動されたかどうかをチェックできます。

```
SELECT DB_PROPERTY ( 'WriteChecksum' );
```

このクエリは、ページがディスクに書き込まれるときにのみチェックサムが有効になっている場合は ON を返し (グローバルチェックサムまたは書き込みチェックサムが有効になっているため)、それ以外の場合は OFF を返します。

次の項を参照してください。


- 「-wc dbeng12/dbsrv12 サーバーオプション」288 ページ
- 「-wc dbeng12/dbsrv12 データベースオプション」317 ページ
- 「START DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「CREATE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「ALTER DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- WriteChecksum データベースプロパティ 810 ページ


### 書き込みチェックサムの自動作成

次のような場合には、データベース作成時に指定したグローバルチェックサム設定に関係なく、データベースの書き込みチェックサムが有効になります。

- **重要なページ**　グローバルチェックサムが有効かどうかに関係なく、データベースサーバーはすべてのデータベース内の重要なデータベースページの書き込みチェックサムを計算します。計算されたチェックサムは、オフラインの破損を検出するために使用します。チェックサムによって、重要なページの破損が原因で他のデータが破損するのを防ぐことができます。データベースサーバーがチェックサムを計算するので、チェックサムが有効に指定されていないデータベースが壊れた場合、データベースサーバーは致命的なエラーとともに停止します。

  また、グローバルチェックサムが有効に指定されていないデータベースを検証した場合でも、そのデータベースに重要なページの破損があると、dbvalid はチェックサム検証に関する警告を返します。

- **Windows Mobile データベース**　Windows Mobile 上で動作しているデータベースに対しては、データベースファイルの破損を早期に検出するために、書き込みチェックサムが自動的に有効になります。

- **一部のストレージメディアで動作しているデータベース**　信頼性の低いストレージメディア (ネットワークドライブやリムーバブルドライブなど) 上でデータベースが実行されている場合、そのデータベースに対する書き込みチェックサムが自動的に有効になります。データベースがそのようなデバイス上にある間は書き込みチェックサムが有効なままになり、ページが書き込まれるたびに、ページにチェックサムが追加されます。信頼性の高いストレージデバイスにデータベースを移動すると、チェックサムが追加されたページがデータベースサーバーのキャッシュに入ったときに、そのページに対してチェックサムの検証が実行されます。

次の項を参照してください。

- 「CREATE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- Checksum データベースプロパティ 785 ページ

### チェックサムの検証

グローバルチェックサムまたは書き込みチェックサムを使用するデータベースのディスクページを検証できます。データベースでグローバルチェックサムが有効になっている場合は、すべてのデータベースページが検証されます。データベースで書き込みチェックサムのみが使用されている場合は、チェックサムを含むページのみが検証されます。チェックサムの検証には、DBA または VALIDATE 権限が必要です。

チェックサムを有効にしたデータベースの場合、チェックサムは各データベースページで計算され、ページがディスクに書き込まれる際にその値が格納されます。検証ユーティリティ (dbvalid)、VALIDATE 文、sa_validate システムプロシージャー、または Sybase Central の**データベース検証ウィザード**を使用してチェックサムの検証を実行できます。この検証では、ディスクからのデータベースページの読み込みとそのページに対するチェックサムの計算が行われます。計算されたチェックサムが格納されているページのチェックサムと一致しない場合、ディスク上で、または書き込み中に、そのページが変更または破壊されています。1 つまたは複数のページが破壊されている場合、エラーが返されて無効なページに関する情報がデータベースサーバーメッセージウィンドウに表示されます。

次の項を参照してください。

- 「VALIDATE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「検証ユーティリティ (dbvalid)」1064 ページ
- 「sa_validate システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』

## テーブルの検証

**♦ テーブルの妥当性を検証するには、次の手順に従います (Sybase Central の場合)。**

1. **SQL Anywhere 12** プラグインを使用して、DBA または VALIDATE 権限のあるユーザーとして、データベースに接続します。
2. 左ウィンドウ枠で、**[テーブル]** をダブルクリックします。
3. テーブルを右クリックし、**[検証]** をクリックします。
4. **[OK]** をクリックします。

**♦ テーブルの妥当性を検証するには、次の手順に従います (SQL の場合)。**

1. DBA 権限または VALIDATE 権限のあるユーザーとしてデータベースに接続します。
2. VALIDATE TABLE 文を実行します。

```
VALIDATE TABLE table-name;
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**注意**

- エラーがレポートされた場合は、テーブル上のすべてのインデックスとキーをいったん削除してから再作成できます。テーブルに対する外部キーも再作成する必要があります。
- 疑わしいインデックスがある場合は、ALTER INDEX...REBUILD 文を実行して、破損したインデックスを再構築できます。「ALTER INDEX 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。
- VALIDATE TABLE によってレポートされるエラーを解決するには、データベース全体をいったんアンロードし、再ロードする方法もあります。アンロード処理で、破損した可能性のあるインデックスがデータの順序付けに使用されないように、dbunload の -u オプションを使用してください。

# スケジュールとイベントの使用によるタスクの自動化

データベース管理タスクの多くは、体系的に実行すると効果的です。たとえば、定期的なバックアップ手順はデータベース管理手順の重要な部分です。

データベースに「イベント」を追加し、イベントのスケジュールを設定することによって SQL Anywhere のルーチンタスクを自動化できます。スケジュールに設定されている時刻になると、いつでも「「イベントハンドラー」」と呼ばれる一連のアクションがデータベースサーバーによって実行されます。

また、データベース管理では、ある状態が発生したときにアクションを実行することも必要です。たとえば、トランザクションログが格納されているディスクの空き領域が少なくなってきたときには、適切な処置を行うよう、システム管理者に電子メールで通知することが考えられます。これらのタスクも各「システムイベント」に対してイベントハンドラーを定義することによって自動化できます。

**質問と回答**

| 質問 | 参照先 |
|---|---|
| スケジュールとは？ | 「スケジュール」 |
| イベントとは？ | 「イベント」 |
| システムイベントとは？ | 「システムイベント」 |
| イベントハンドラーとは？ | 「イベントハンドラー」 |
| イベントハンドラーのデバッグ方法は？ | 「イベントハンドラーの開発」 |

| 質問 | 参照先 |
| --- | --- |
| データベースサーバーがスケジュールを使用してイベントハンドラーをトリガーする仕組みは？ | 「データベースサーバーによるスケジュールされたイベントのチェック」 |
| 定期バックアップをスケジュールする方法は？ | 「スケジュール」 |
| データベースサーバーがイベントハンドラーをトリガーするのに使用できるシステムイベントの種類は？ | 「システムイベント」<br><br>「CREATE EVENT 文」『SQL Anywhere サーバー SQL リファレンス』 |
| イベントハンドラーの実行に使用される接続は？ | 「イベントハンドラーの実行」 |
| イベントハンドラーがコンテキスト情報を得る仕組みは？ | 「イベントハンドラーの開発」<br><br>「EVENT_PARAMETER 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』 |

# イベント

データベースにイベントを追加し、イベントのスケジュールを設定することによって SQL Anywhere のルーチンタスクを自動化できます。SQL Anywhere では、3 種類のイベントをサポートしています。

- **スケジュールされたイベント**　スケジュールに関連付けられており、指定の時間に実行されます。「スケジュール」1124 ページを参照してください。
- **システムイベント**　データベースサーバーによって追跡される特定のタイプの条件に関連付けられています。「システムイベント」1126 ページを参照してください。
- **手動イベント**　TRIGGER EVENT 文を使用して明示的に起動されます。「イベントハンドラーのトリガー」1135 ページを参照してください。

イベントハンドラーが実行されるたびに、エラーが発生しなかった場合、COMMIT が発生します。エラーが生成された場合は、ROLLBACK が発生します。

# スケジュール

アクティビティをスケジュールすると、事前に設定した時刻に一連のアクションを確実に実行できます。スケジュール情報とイベントハンドラーはいずれもデータベース自体に格納されます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

通常は必要ありませんが、1 つの名前付きイベントに 2 つ以上のスケジュールを関連付けることによって複雑なスケジュールを定義できます。たとえば、営業時間が曜日によって変わるような販売店で、営業時間中 1 時間に 1 回イベントを発生させることができます。それぞれ別のスケジュールを持つ複数のイベントを定義して、共通のストアドプロシージャーを呼び出すことで同じような効果が得られます。

イベントをスケジュールするとき、英語の曜日をフルネーム (Monday、Tuesday など) で使用することも、省略形 (Mon、Tue など) で使用することもできます。英語以外の言語で稼働するサーバーで曜日名を認識する必要がある場合は、フルネームの英語の曜日を使用する必要があります。

イベントは、現在ミラーサーバーとして動作しているサーバーでは起動しません。また、ミラーリングイベントを監視サーバー上で実行するよう定義することはできません。なぜなら、イベントは定義されているデータベースのコンテキストでのみ実行されますが、監視サーバーはミラーリングされているデータベースのコピーを使用しないからです。

次に便利なスケジュールの例を示します。

## 参照

- 「CREATE EVENT 文」『SQL Anywhere サーバー SQL リファレンス』

## 例

インクリメンタルバックアップを毎日午前 1 時に実行します。

```
CREATE EVENT IncrementalBackup
SCHEDULE
 START TIME '1:00 AM' EVERY 24 HOURS
HANDLER
BEGIN
 BACKUP DATABASE DIRECTORY 'c:¥¥backup'
 TRANSACTION LOG ONLY
 TRANSACTION LOG RENAME MATCH
END;
```

終業時に受注の要約を作成します。

```
CREATE EVENT Summarize
SCHEDULE
 START TIME '6:00 pm'
 ON ( 'Monday', 'Tuesday', 'Wednesday', 'Thursday',
  'Friday' )
HANDLER
 BEGIN
 INSERT INTO OrderSummary
  SELECT CURRENT DATE,
     COUNT( * ),
     SUM( amount )
  FROM Orders
  WHERE date_ordered = current date
END;
```

## スケジュール定義

柔軟に設定を可能にするために、スケジュール定義にはいくつかの構成要素が用意されています。

- **名前**　各スケジュール定義には名前があります。2 つ以上のスケジュールを 1 つのイベントに割り当てることができます。これは複雑なスケジュールを設計するのに便利です。
- **起動時刻**　イベントの起動時刻を定義できます。これは、イベントの実行が開始する時刻です。
- **範囲**　起動時刻の代わりとして、イベントがアクティブになる時刻の範囲を指定できます。イベントは、指定した起動時刻と終了時刻の間に発生します。頻度は指定した周期で決定します。
- **周期**　各スケジュールには周期を定義できます。イベントは、何日または何曜日ごとの何時何分何秒ごとという形で指定できる頻度でトリガーされます。反復イベントには、**EVERY** または **ON** 句が含まれています。

イベントのスケジュールは、CREATE EVENT 文に定義できるほか、スケジュール作成ウィザードを使用して定義することもできます。

イベントの作成時にスケジュールを追加する方法については、「CREATE EVENT 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

**♦ イベントのスケジュールの作成 (Sybase Central の場合)**

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。
2. **[イベント]** をダブルクリックします。
3. スケジュールを作成するイベントをダブルクリックします。
4. **[スケジュール]** タブをクリックします。
5. **[ファイル]** » **[新規]** » **[スケジュール]** をクリックします。
6. **スケジュール作成ウィザード**の指示に従います。

## システムイベント

SQL Anywhere は、いくつかのシステムイベントを追跡します。各システムイベントが提供するフックに一連のアクションをハングすることができます。データベースサーバーはイベントを追跡し、システムイベントが定義された「トリガー条件」を満たしたときに (イベントハンドラーに定義された) アクションを実行します。

トリガー条件の詳細については、「イベントのトリガー条件」1129 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

イベントハンドラーを定義して、選択したシステムイベントが発生し、定義したトリガー条件を満たしたときに実行されるようにします。このようにしておくことで、データのセキュリティが向上し、管理が容易になります。イベントハンドラーの動作は、実行中にエラーが検出されなければコミットされ、エラーが検出された場合はロールバックされます。

使用可能なシステムイベントは次のとおりです。

- **BackupEnd**　BackupEnd イベントタイプを使用すると、バックアップ終了時にアクションを実行できます。

- **接続イベント**　接続が確立されたとき (Connect) または接続できなかったとき (ConnectFailed)。これらのイベントはセキュリティの目的で使用できます。イベントハンドラーに接続する代わりに、ログインプロシージャーを使用することもできます。「login_procedure オプション」636 ページを参照してください。

- **DatabaseStart**　DatabaseStart イベントタイプを使用すると、データベース起動時にアクションを実行できます。

- **Deadlock**　Deadlock イベントを使用すると、デッドロック発生時にアクションを実行できます。イベントハンドラーでは、sa_report_deadlocks プロシージャーを使用して、デッドロックが発生するに至った状況に関する情報を取得できます。Deadlock イベントを使用するときは、データベースサーバーがデッドロック情報を取得するように設定する必要があります。これを行うには、log_deadlocks オプションを On にし、sa_server_option または -zl サーバーオプションを使用して RememberLastStatement 機能を有効にします。

  Deadlock イベントは、接続デッドロックとスレッドデッドロックの発生時に起動します。デッドロックイベントは、sa_report_deadlocks システムプロシージャーによって取得できるもの以上の情報は提供しません。しかし、このイベントを使用すると、デッドロックにすぐに対処できます。データベースサーバーに保持されるデッドロック関連の情報は量が限られているため、迅速な対応が重要なこともあります。次の項を参照してください。

  ○「sa_report_deadlocks システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
  ○「log_deadlocks オプション」634 ページ
  ○「デッドロック」『SQL Anywhere サーバー SQL の使用法』

- **Disconnect**　Disconnect イベントを使用すると、ユーザーまたはアプリケーションの切断時にアクションを実行できます。

- **ディスクの空き領域**　データベースファイル (DBDiskSpace)、ログファイル (LogDiskSpace)、テンポラリファイル (TempDiskSpace) を格納しているデバイスの使用可能なディスク領域を追跡します。このシステムイベントは、Windows Mobile では使用できません。

  ディスク領域イベントを使用すると、ディスク領域が不足したときに管理者に警告することができます。

  データベースサーバーの起動時に -fc オプションを指定して、データベースサーバーでファイルシステムがいっぱいになった場合のコールバック関数を実装できます。「-fc dbeng12/dbsrv12 サーバーオプション」229 ページを参照してください。

- **ファイルサイズ**　ファイルが指定したサイズに達したとき。これはデータベースファイル (GrowDB)、トランザクションログ (GrowLog)、テンポラリファイル (GrowTemp) に使用できます。

  ファイルサイズイベントを使用すると、データベース上での異常なアクションを追跡したり、バルクオペレーションをモニターすることができます。

- **GlobalAutoincrement**　GLOBAL AUTOINCREMENT で定義されたカラムの残りの値がこの範囲の 1% を下回ると、GlobalAutoincrement イベントが起動します。これは、このイベント用のパラメーターとして指定された残りの値のテーブルと数字に基づいて、global_database_id オプション用の新しい値を要求するのに使用できます。このイベントにおけるテーブルの残りの値を取得するには、EVENT_PARAMETER 関数を使用し、RemainingValues パラメーターと TableName パラメーターを指定します。RemainingValues は、そのカラム用に生成できる残りの値の数を返します。TableName は、範囲の終わりに近づいている GLOBAL AUTOINCREMENT カラムがあるテーブルを返します。「EVENT_PARAMETER 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

- **RAISERROR エラー**　RAISERROR イベントタイプを使用すると、RAISERROR 文が実行されたときにアクションを実行できます。RAISERROR 文で使用するエラー番号は、EVENT_CONDITION 関数 (たとえば、EVENT_CONDITION( 'ErrorNumber' )) を使用してイベントハンドラー内に定義できます。

- **アイドル時間**　データベースサーバーが指定した時間アイドル状態にあったとき (ServerIdle)。このイベントタイプを使用すると、定型の管理操作をアクセスの少ない時間に行えます。

- **データベースミラーリング**　プライマリサーバーからミラーサーバーまたは監視サーバーへの接続が失われると、MirrorServerDisconnect イベントが起動します。接続が失われたサーバーの名前を取得するには、EVENT_PARAMETER 関数を使用し、MirrorServerName パラメーターを指定します。「EVENT_PARAMETER 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

  サーバーがデータベースの所有権を取得すると、MirrorFailover イベントが起動します。たとえば、初めてサーバーが起動され、データベースを所有する必要があると判断した場合に起動します。また、前回ミラーとして動作していたサーバーが、プライマリサーバーがダウンしており、監視サーバーに確認した結果、所有権を取得する必要があると判断した場合にも起動します。

  イベントは、現在ミラーサーバーとして動作しているサーバー上では起動しません。これは、データベースのコピーが依然として起動されている状態であるからです。また、ミラーリングイベントを監視サーバー上で実行するよう定義することはできません。これは、イベントは定義されているデータベースのコンテキストでのみ実行されますが、監視サーバーはミラーリングされているデータベースのコピーを使用しないからです。「データベースミラーリングにおけるシステムイベント」1173 ページを参照してください。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## イベントのトリガー条件

各イベント定義には対応するシステムイベントがあります。また、イベント定義は 1 つまたは複数のトリガー条件を持ちます。システムイベントに対するトリガー条件が満たされるとイベントハンドラーがトリガーされます。

トリガー条件は CREATE EVENT 文の WHERE 句に含まれていて、AND キーワードを使用して結合できます。各トリガー条件は次のフォームで定義します。

**event_condition**( *condition-name* ) *comparison-operator value*

*condition-name* 引数は、さまざまなイベントタイプに対応できるようにあらかじめ設定されている文字列から 1 つを選択します。たとえば、**DBSize** (メガバイト単位のデータベースファイルサイズ) を使用して **GrowDB** システムイベントに適したトリガー条件を構築することができます。データベースサーバーは、条件名とイベントタイプの対応をチェックしません。イベントタイプのコンテキストで条件に意味があるかどうかを確認する必要があります。

### 例

● トランザクションログのサイズを 10 MB に制限します。

```
CREATE EVENT LogLimit
TYPE GrowLog
WHERE event_condition( 'LogSize' ) > 10
HANDLER
BEGIN
  IF EVENT_PARAMETER( 'NumActive' ) = 1 THEN
    BACKUP DATABASE
    DIRECTORY 'c:¥¥logs'
    TRANSACTION LOG ONLY
    TRANSACTION LOG RENAME MATCH;
  END IF;
END;
```

●データベースファイルを格納しているデバイスの空き領域が 10 % を下回ると管理者に通知しますが、ハンドラーは 5 分間 (300 秒) に 2 回以上実行しません。

```
CREATE EVENT LowDBSpace
TYPE DBDiskSpace
WHERE event_condition( 'DBFreePercent' ) < 10
AND event_condition( 'Interval' ) >= 300
HANDLER
BEGIN
 CALL xp_sendmail( recipient='DBAdmin',
    subject='Low disk space',
    "message"='Database free disk space '
    || EVENT_PARAMETER( 'DBFreeSpace' ) );
END;
```

●データベースに侵入しようとする者を発見すると、管理者に通知します。

```
CREATE EVENT SecurityCheck
TYPE ConnectFailed
HANDLER
BEGIN
```

```
 DECLARE num_failures INT;
 DECLARE mins INT;
 INSERT INTO FailedConnections( log_time )
 VALUES ( CURRENT TIMESTAMP );

 SELECT COUNT( * ) INTO num_failures
 FROM FailedConnections
 WHERE log_time >= DATEADD( minute, -5,
  current timestamp );
 IF( num_failures >= 3 ) THEN
  SELECT DATEDIFF( minute, last_notification,
   current timestamp ) INTO mins
  FROM Notification;
  IF( mins > 30 ) THEN
   UPDATE Notification
   SET last_notification = current timestamp;
   CALL xp_sendmail( recipient='DBAdmin',
      subject='Security Check', "message"=
       'over 3 failed connections in last 5 minutes' )
  END IF
 END IF
END;
```

- サーバーが 10 分間以上アイドル状態にあると、処理を実行します。1 時間に 2 回以上は実行しません。

```
CREATE EVENT Soak
TYPE ServerIdle
WHERE event_condition( 'IdleTime' ) >= 600
AND event_condition( 'Interval' ) >= 3600
HANDLER
BEGIN
 MESSAGE ' Insert your code here ... '
END;
```

# イベントハンドラー

イベントハンドラーは、イベントをトリガーするアクションとは別の接続上で実行されます。そのため、クライアントアプリケーションに影響することはありません。イベントハンドラーは、イベントの作成者のパーミッションで実行されます。

## イベントハンドラーの開発

イベントハンドラーは、スケジュールされたイベント用か、システムイベント処理用かにかかわらず、複合文を含んでいて、多くの点でストアドプロシージャーに似ています。ループや条件付き実行などを追加することができます。また、SQL Anywhere デバッガーを使ってイベントハンドラーをデバッグすることができます。

各イベントハンドラーの実行後、エラーが発生しない場合、COMMIT が発生します。エラーが発生した場合は、ROLLBACK が発生します。

### イベントハンドラーのためのコンテキスト情報

ストアドプロシージャーとは異なり、イベントハンドラーには引数がありません。
EVENT_PARAMETER 関数を使用して、イベントがトリガーされたコンテキストに関する情報に

アクセスできます。返される情報には、イベントがトリガーされた接続 ID とユーザー ID、イベント名、実行回数が含まれます。「EVENT_PARAMETER 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

### イベントハンドラーのテスト

開発中は、好きなときにイベントハンドラーをトリガーできた方が便利です。TRIGGER EVENT 文を使うと、トリガー条件やスケジュールした時刻に関係なく、明示的にイベントを実行できます。ただし、無効なイベントハンドラーを TRIGGER EVENT によって実行することはできません。「TRIGGER EVENT 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

運用データベース上でイベントハンドラーを開発するのはよいことではありませんが、Sybase Central から、または明示的に ALTER EVENT 文を使ってイベントハンドラーを無効にすることができます。

### コードの共有

複数のイベントを処理するアクションを 1 つにまとめておくと便利です。たとえば、データベースファイルまたはログファイルを格納しているデバイスのディスク領域が少なくなってきたときに、通知アクションを実行することができます。これを実行するには、ストアドプロシージャーを作成し、各イベントハンドラーの本文から呼び出します。このとき、必要なコンテキスト情報をパラメーターとしてプロシージャーに渡します。

### イベントハンドラーのデバッグ

イベントハンドラーのデバッグは、ストアドプロシージャーのデバッグによく似ています。イベントハンドラーは、イベントリストに表示されます。「イベントハンドラーのデバッグ」1136 ページを参照してください。

### イベントハンドラーを隠す

SET HIDDEN 句を使用して、イベントハンドラーの定義を隠すことができます。SET HIDDEN 句を指定すると、ISYSEVENT システムテーブルの action カラムに格納されているイベントハンドラーの定義が永続的に難読化されます。「ALTER EVENT 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

### アクティブなイベントの制限

また、NumActive イベントパラメーターを使用して、現在アクティブになっている特定のイベントハンドラーのインスタンスの数を判断できます。この関数は、一定時間に 1 つのイベントハンドラーで 1 つのインスタンスだけを実行させるように制限する場合に役立ちます。

NumActive イベントパラメーターの詳細については、「EVENT_PARAMETER 関数 [システム]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

## スケジュールとイベントの内部

この項ではデータベースサーバーがスケジュールとイベント定義を処理する仕組みについて説明します。

## データベースサーバーによるシステムイベントのチェック

システムイベントは「イベントタイプ」によって分類されます。イベントタイプは、CREATE EVENT 文の中で直接指定するか、Sybase Central を使って指定します。イベントタイプには 2 つの種類があります。

- **アクティブイベントタイプ**　イベントタイプには、データベースサーバー自体のアクションの結果であるものがあります。こうしたアクティブなイベントタイプには、データベースファイルサイズ、さまざまなデータベースアクションの開始時、終了時 (BackupEnd など)、RAISERROR などが含まれます。

  データベースサーバーは、アクションを実行するときに、WHERE 句に定義されたトリガー条件が満たされているかどうかをチェックし、条件が満たされていればイベントタイプに対して定義されたイベントをトリガーします。

- **ポーリングイベントタイプ**　ディスクの空き領域 (DBDiskSpace など) や IdleTime などのイベントタイプは、データベースのアクションだけでトリガーされません。

  このタイプのイベントに対して、データベースサーバーは 30 秒ごとにポーリングします。ポーリングはデータベースの開始後、約 30 秒後から開始されます。

  IdleTime イベントタイプの場合、データベースサーバーはサーバーが 30 秒間アイドル状態にあったかどうかをチェックします。その間まったく要求が開始されず、現在アクティブな要求もなければ、秒単位のアイドルチェック間隔時間をアイドル時間の合計に追加します。そうでない場合はアイドル時間の合計が 0 にリセットされます。したがって、IdleTime の値は、常に 30 秒の倍数になります。IdleTime がトリガー条件に指定した間隔より長くなると、IdleTime に関連付けられたイベントハンドラーが起動します。

## データベースサーバーによるスケジュールされたイベントのチェック

イベントのスケジュール時刻の計算は、データベースサーバーの起動時と、スケジュールされた各イベントハンドラーの完了時に行われます。

次回のスケジュール時刻の計算は、スケジュール定義に指定された増分に基づいて、増分を前回の起動時刻に追加することで行われます。指定した増分よりイベントハンドラーの実行時間が長くなり、現在の処理が終わらないうちに次回のスケジュール時刻が来てしまう場合、データベースサーバーは、現在の処理の後に次のスケジュール時刻がくるように増分します。

たとえば、実行に 65 分かかるイベントハンドラーが 9 時 ～ 5 時の間の 1 時間ごとに起動するように要求された場合、実際には 9 時、11 時、1 時と 2 時間ごとに実行されます。

次回の実行まで待機時間を設ける処理を 9 時 ～ 5 時の間で実行するには、各実行の合間に WAITFOR 文を使って、指定した完了時間が経過するまでループするようにハンドラーを定義できます。

データベースサーバーを断続的に実行していて、スケジュール時刻にデータベースサーバーが実行中でない場合、イベントハンドラーが起動時に実行されることはありません。その代わりに、次のスケジュール時刻は起動時に計算されます。たとえば、毎晩 1 時にバックアップを実行する

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

ようにスケジュールしていても、終業時にはいつもデータベースサーバーを停止している場合、バックアップが実行されることはありません。

次にスケジュールされているイベント実行が 1 時間以上後の場合、データベースサーバーは時間単位で次のスケジュール時間を計算します。これにより、夏時間の開始または終了のためにシステムクロックが調整されたときに、イベントが予定どおりに起動されます。

## イベントハンドラーの実行

イベントハンドラーがトリガーされると、一時的に内部接続が確立され、その上でイベントハンドラーが実行されます。ハンドラーは、そのハンドラーがトリガーされるに至った接続で実行されるわけではありません。このため、クライアントアプリケーションとの対話に使用される MESSAGE...TO CLIENT などの文は、イベントハンドラー内では意味を持ちません。同様に、結果セットを返す文は使用できません。

ハンドラーが実行される一時的な接続は、ライセンス契約の接続制限には数えられません。login_procedure に指定したプロシージャーはイベント接続では実行されません。

イベントの作成には DBA 権限が必要です。また、イベントは作成者のパーミッションで実行されます。DBA 権限を持たないでイベントハンドラーを実行する場合、作成者のパーミッションで実行されるストアドプロシージャーのように、ハンドラー内からプロシージャーを呼び出すことができます。

イベントエラーが発生すると、データベースサーバーメッセージログに記録されます。

**注意**
イベントハンドラー内のトランザクションは、実行中にエラーが検出されなかった場合はコミットされ、エラーが検出された場合はロールバックされます。「例外処理とアトミックな複合文」『SQL Anywhere サーバー SQL の使用法』を参照してください。

# イベント処理タスク

この項では、イベントを使用したタスクの自動化に関連するタスクの手順について説明します。

## データベースへのイベントの追加

♦ **データベースにイベントを追加するには、次の手順に従います (Sybase Central の場合)。**

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。
2. 左ウィンドウ枠で **[イベント]** を右クリックし、**[新規]** » **[イベント]** をクリックします。
3. **イベント作成ウィザード**の指示に従います。

イベントのオプションについては、他のタスクの項で詳しく説明します。

**♦ データベースにイベントを追加するには、次の手順に従います (SQL の場合)。**

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. CREATE EVENT 文を実行します。

   CREATE EVENT 文には、作成するイベントに応じて多くのオプションがあります。詳細については各タスクの中で説明しています。

   「CREATE EVENT 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

## 手動トリガーイベント

トリガーするスケジュールもシステムイベントも設定しないでイベントハンドラーを作成した場合、それは手動でトリガーしたときのみ実行されます。

イベントハンドラーを開発する場合、スケジュールやシステムイベントを追加して、後からイベントのトリガーを制御できます。これには Sybase Central または ALTER EVENT 文を使用します。

**♦ データベースに手動トリガーイベントを追加するには、次の手順に従います (Sybase Central の場合)。**

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。
2. 左ウィンドウ枠で **[イベント]** を右クリックし、**[新規]** » **[イベント]** をクリックします。
3. **[新しいイベントの名前を指定してください。]** フィールドにイベントの名前を入力し、**[次へ]** をクリックします。
4. **[手動]** をクリックし、**[次へ]** をクリックします。
5. **[このイベントを有効にする]** » **[すべてのデータベースで実行]** をクリックしてから、**[次へ]** をクリックします。
6. イベントを説明するコメントを入力し、**[完了]** をクリックします。
7. **[SQL]** ウィンドウ枠で、イベントで使用する SQL 文を入力します。
8. **[ファイル]** » **[保存]** をクリックします。

**♦ データベースに手動トリガーイベントを追加するには、次の手順に従います (SQL の場合)。**

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. スケジュールや WHERE 句なしで CREATE EVENT 文を実行します。CREATE EVENT の構文は、次のとおりです。

```
CREATE EVENT event-name
HANDLER
BEGIN
... //event handler
END
```

**参照**

- 「イベントハンドラーのトリガー」1135 ページ
- 「ALTER EVENT 文」『SQL Anywhere サーバー SQL リファレンス』

## イベントハンドラーのトリガー

すべてのイベントハンドラーはスケジュールやシステムイベントによって起動されるほか、手動でトリガーすることができます。開発中や、運用環境におけるいくつかのイベントに関しては、手動によるイベントのトリガーが便利です。たとえば、毎月の売り上げレポートをスケジュールしている場合、月末でなくても売り上げレポートが必要になることがあります。

イベントハンドラーの開発については、「イベントハンドラーの開発」1130 ページを参照してください。

**♦ イベントハンドラーをトリガーするには、次の手順に従います (Sybase Central の場合)。**

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。
2. 左ウィンドウ枠で、**[イベント]** をダブルクリックします。
3. イベントを右クリックして、**[トリガー]** をクリックします。

   イベントは、有効にしておかなければトリガーすることはできません。イベントを有効にするには、イベントを右クリックして **[有効化]** をクリックします。
4. **[パラメーター]** フィールドに、イベントで使用するパラメーターをカンマで区切って入力します。次に例を示します。

   ```
   parameter=value,parameter=value
   ```
5. **[OK]** をクリックします。

**♦ イベントハンドラーをトリガーするには、次の手順に従います (SQL の場合)。**

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. イベントの名前を指定して、TRIGGER EVENT 文を実行します。次に例を示します。

   ```
   TRIGGER EVENT sales_report_event;
   ```

「TRIGGER EVENT 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

## イベントハンドラーのデバッグ

♦ **イベントハンドラーをデバッグするには、次の手順に従います (Sybase Central の場合)。**

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。

   DBA 権限のあるユーザーとして、データベースに接続します。

2. **[モード]** » **[デバッグ]** をクリックします。
3. 左ウィンドウ枠で、**[イベント]** をダブルクリックします。
4. デバッグするイベントをダブルクリックします。
5. 右ウィンドウ枠の **[SQL]** タブで、[F9] キーを押してブレークポイントを設定します。
6. Interactive SQL または他のアプリケーションから、TRIGGER EVENT 文を使用してイベントハンドラーをトリガーします。
7. 設定したブレークポイントで実行が停止します。

**参照**

- 「イベントハンドラーの開発」1130 ページ
- 「SQL Anywhere のデバッガー」『SQL Anywhere サーバー SQL の使用法』

## イベントハンドラーを隠す

セキュリティ向上のために、ALTER EVENT 文を使用してイベントハンドラーの定義を隠すことができます。この文を使用すると、ISYSEVENT システムテーブルの action カラムに格納されているイベントハンドラーの定義が難読化されます。

♦ **イベントハンドラーを隠すには、次の手順に従います (SQL の場合)。**

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. ALTER EVENT *event-name* SET HIDDEN 文を実行します。*event-name* には、ハンドラーを隠すイベントの名前を入力します。

**参照**

- 「ALTER EVENT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「SYSEVENT システムビュー」『SQL Anywhere サーバー SQL リファレンス』

# SQL Anywhere の高可用性

SQL Anywhere では、高可用性を実現するためにデータベースミラーリングと Veritas Cluster Server をサポートしています。これらの両方の構成によって、現在データベースを実行している

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

データベースサーバーが使用できなくなった場合に、データベースを実行できる代替データベースサーバーが提供されます。

負荷を分散させる場合は、読み込み専用のスケールアウトを使用することが適切な解決法であるかどうかも検討してください。読み込み専用のスケールアウトは、データベースミラーリングでのみ使用できます。「SQL Anywhere の読み込み専用のスケールアウト」1188 ページと「データベースミラーリングシステムの読み込み専用スケールアウトシステム」1203 ページを参照してください。

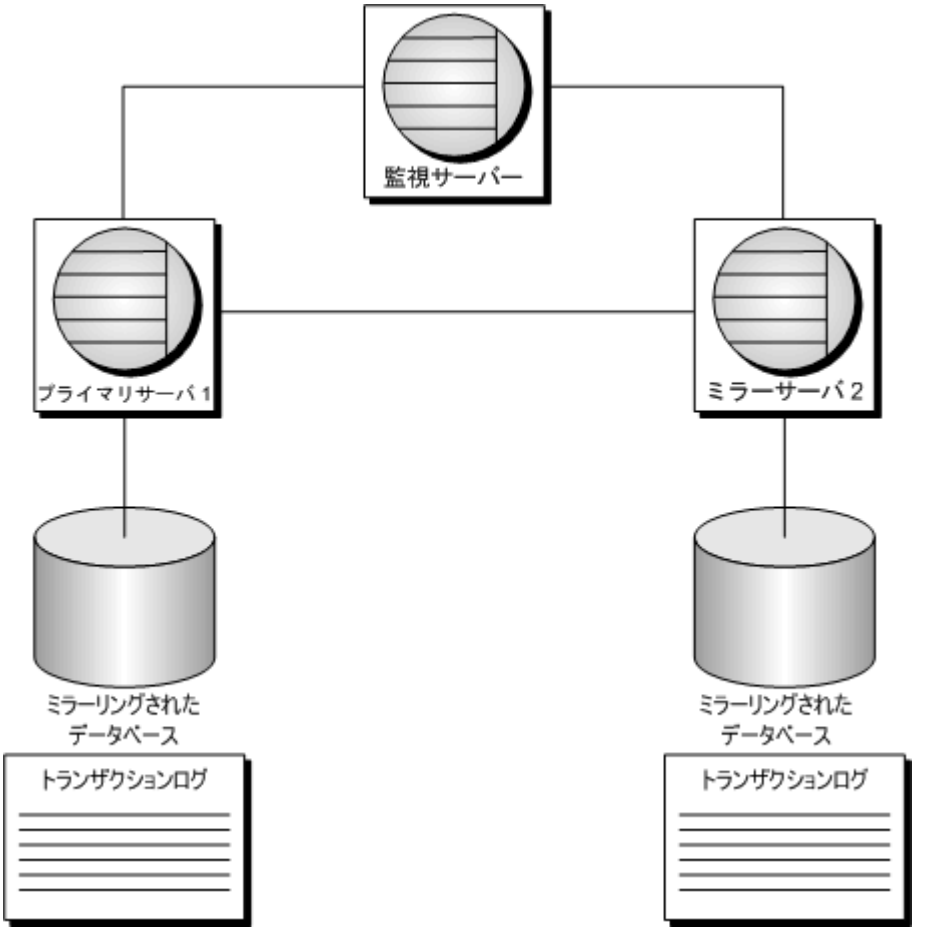
Web アプリケーションをホストするために高可用性かつ読み込み専用のスケールアウトを使用する方法の詳細については、「SQL Anywhere Web サービスの高可用性とスケールアウトソリューション」1205 ページを参照してください。

# データベースミラーリング

**注意**
データベースミラーリングには別途ライセンスが必要です。「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

「データベースミラーリング」とは、異なるコンピューター上で実行されている 2 つまたは 3 つのデータベースサーバーが連携してデータベースやトランザクションログファイルのコピーを保持する構成です。

「プライマリサーバー」と「ミラーサーバー」は、それぞれがデータベースファイルとトランザクションログファイルのコピーを保持し、「監視サーバー」と呼ばれる第 3 のサーバーが、必要に応じて、前記 2 つのサーバーのどちらにデータベースの所有権を持たせるかを決定します。監視サーバーはデータベースのコピーを保持しません。この 3 種類のデータベースサーバー (プライマリ、ミラー、監視の各サーバー) による構成は「ミラーリングシステム」と呼ばれ、プライマリサーバーとミラーサーバーは併せて「パートナー」サーバーと呼ばれます。




クライアントがデータベースにアクセスする場合は、プライマリサーバーに接続します。データベースに対するあらゆる変更は、プライマリサーバー上のトランザクションログファイルに記録されます。変更がコミットされると、トランザクションログページがミラーサーバーに送信され、データベースのミラーコピーに適用されます。サーバーがミラーサーバーとして動作している間は、そのミラーサーバーにあるデータベースのコピーには読み込み専用モードでのみアクセスできます。「ミラーサーバーで実行されているデータベースへの読み込み専用アクセスの設定」1165 ページを参照してください。

プライマリサーバーがハードウェアやソフトウェアの障害により使用できなくなった場合、ミラーサーバーが監視サーバーとネゴシエートしてデータベースの所有権を取得し、プライマリサーバーのロールを引き受けます。所有権の譲渡、すなわち「ロールの切り替え」を実施するには、ダウンしていないパートナーサーバーと監視サーバーは、ロールの切り替えを行おうとする時点で、ミラーが最新で同期された状態にあることを合意する必要があります。それまで元のプライマリサーバーに接続されていたクライアントはすべて切断され、コミットされていないトランザクションはすべて失われます。クライアントがデータベースへのアクセスを続行するには、新しいプライマリサーバー上のデータベースに接続し直す必要があります。元のプライマリサーバーは、再び使用可能になると、ミラーサーバーのロールを引き受けます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

データベースサーバーのデータベースサーバーメッセージウィンドウには、起動時に、そのサーバーが引き受けているロールと起動プロセスの進捗状況を示すステータスメッセージが表示されます。また、ミラーリングシステムを構成する他の 1 つ以上のサーバーが失われたことが原因でデータベースの再起動が必要になった場合、またはロールがミラーからプライマリに変更された場合に、メッセージが表示されます。

ミラーリングシステムを構成するいずれかのサーバーでアサーションが失敗すると、そのサーバーはデータベースサーバーメッセージログにエラーを出力して終了します。これにより、他のサーバーに障害発生が通知され、適切なアクションが実行されます。

データベースミラーリングにハードウェアやソフトウェアに関する特別な要件はありません。また、各データベースサーバーは、地理的に離れたロケーションで実行されていてかまいません。データベースミラーリングシステムを構成するデータベースサーバーでは、ミラーリングされたデータベースもされていないデータベースも実行できます。また、監視サーバーは、複数のデータベースミラーリングシステムの監視サーバーになることができます。

高可用性環境のデータベースで、互換性がない、トランザクションログが一致しないなどの問題が発生すると、データベースは停止します。問題が発生したデータベースを実行していたデータベースサーバーも、他に実行中のデータベースがないかぎりシャットダウンします。

データベースミラーリングシステムの各データベースのステータスの詳細は、ステータス情報ファイルに格納されます。「ステータス情報ファイル」1145 ページを参照してください。

**注意**
データベースミラーリングは、バックアップとリカバリのプランの代用ではありません。データベースには、バックアップとリカバリの手段を必ず実装してください。「データベースミラーリングとバックアップ」1174 ページと「バックアップとデータリカバリ」1071 ページを参照してください。

SQL Anywhere のアップグレード、またはデータベースのミラーリングに使用しているデータベースの再構築については、「データベースミラーリングシステムでの SQL Anywhere ソフトウェアとデータベースのアップグレード」『SQL Anywhere 12 変更点とアップグレード』を参照してください。

## クォーラム

サーバーがプライマリサーバーのロールを引き受けるには、「クォーラム」が必要です。これは、他のサーバーの少なくともどちらか 1 つがデータベースの所有に合意していることを意味します。ミラーサーバーが使用できなくなった場合でも、プライマリサーバーと監視サーバーが接続されていれば、プライマリサーバーは引き続きデータベースへのアクセスを提供します。プライマリサーバーがクォーラムを失った場合、プライマリサーバーはデータベースへのアクセスを許可できなくなります。その時点で、プライマリサーバーは、ミラーリングされているデータベースを停止し、再起動を試行し、クォーラムを再び取得するのを待ってから、データベースをアクセス可能にします。

データベースミラーリングシステムが起動されると、データベースサーバーは起動プロセスを開始してクォーラムを形成し、クライアント接続を受け付けます。このプロセスにおける一般的なイベントの流れは次のとおりです。

1. 監視サーバーは、サーバー 1 とサーバー 2 を待機します。

2. サーバー 1 が監視サーバーまたはサーバー 2 を探します。
3. サーバー 1 が監視サーバーに接続します。
4. サーバー 1 は、プライマリサーバーになるために監視サーバーとネゴシエートします。
5. 監視サーバーとサーバー 1 は、サーバー 1 がプライマリサーバーになることで合意します。
6. サーバー 1 は接続の受け付けを開始します。
7. サーバー 2 は、サーバー 1 と監視サーバーを探します。
8. サーバー 2 が監視サーバーとサーバー 1 に接続します。
9. サーバー 2 はクォーラムを要求します。しかし、サーバー 1 がプライマリなのでクォーラムを取得できず、サーバー 2 はサーバー 1 からのトランザクションの待機状態に入ります。
10. サーバー 1 はトランザクションをサーバー 2 に送信します。

**制限**

データベースミラーリングを使用する場合は次のような制限があります。

- **ネットワークデータベースサーバーが必要**　ミラーリングにはデータベースサーバー間のネットワーク通信が必要なので、ネットワークデータベースサーバー (dbsrv12) を使用する必要があります。パーソナルサーバーは使用できません。
- **LOAD TABLE 文**　データベースミラーリングを使用しているときにベーステーブルに対して LOAD TABLE 文を実行する場合は、文のロギングレベルとして、WITH ROW LOGGING または WITH CONTENT LOGGING を指定する必要があります。これらの句を指定することで、ロードされたデータをトランザクションログに記録して、ミラーリングデータベースにもロードできるようになります。これらの句が指定されていない場合は、エラーがレポートされます。
- **TCP/IP が必要**　ミラーリングサーバー間で許可されるのは TCP/IP 接続だけです。
- **フェールオーバーとスケジュールされたイベント**　スケジュールされたイベントのあるデータベースでフェールオーバーが発生した場合、フェールオーバーはイベントの開始前に完了する必要があります。完了しない場合、イベントは次のスケジュール時刻まで実行されません。ミラーサーバーがプライマリサーバーのロールに切り替わろうとし、切り替えがまだ完了していない場合、スケジュールされたイベントは実行されず、次のスケジュール時刻に実行されます。イベントの実行中にプライマリサーバーがミラーサーバーと監視サーバーへの接続を失った場合、イベント接続と他のすべての接続は切断されます。ミラーサーバーがプライマリサーバーのロールを引き受けた後で実行されるようにイベントがスケジュールされている場合、イベントは新しいプライマリサーバーで実行されます。
- **トランザクションログの制限**　データベースミラーリングを使用している場合、トランザクションログのトランケートは実行できません。これは、トランザクションが失われる可能性があるからです。トランザクションログの名前は、必要に応じて何度でも変更できます。古いトランザクションログを削除する場合は、不要であることを確認したうえで、イベントのスケジュールを使用して削除できます。たとえば、作成されてから 1 週間が過ぎたトランザ

クションログのコピーを削除する毎日実行されるイベントを作成できます。「データベースミラーリングとトランザクションログファイル」1172 ページを参照してください。

- **ミラーリングシステムでの SQL Anywhere Web サーバーの使用**　SQL Anywhere を Web サーバーとしてミラーリングシステムで使用する場合、Web 要求の URL を指定しても、要求が現在のプリマリサーバーに送信されることは保証されません。いずれかのサーバーコンピューターが URL で指定されている場合、そのサーバーが停止すると、要求はタイムアウトします。

### アプリケーション開発時の考慮事項

データベースミラーリングを使用する場合、アプリケーションは、ほとんどのケースでミラーリングされていないデータベースに接続している場合と同じように実行できます。ただし、データベースミラーリングが関係するアプリケーションを開発するうえで、考慮すべき点がいくつかあります。

- データベースに再接続できるクライアントを作成する (たとえば、フェールオーバーが発生した場合、ユーザーがアプリケーションをシャットダウンして再起動する必要が生じることがあります)。

- 非同期モードまたは非同期フルページモードで実行している場合、フェールオーバーが発生してトランザクションがデータベースにコミットされない場合の対処法を決定する必要がある。

- ミラーサーバーがデータベースの所有権を引き継ぐ場合は、不完全なトランザクションをロールバックする必要があるが、トランザクションが長いほど、ロールバックに時間がかかる。フェールオーバーのリカバリ速度は、クライアントの数とロールバックする必要があるトランザクションの長さに影響されます。リカバリ速度が問題になる場合は、できるだけ短いトランザクションを使用するようアプリケーションを設計してください。

### SQL Anywhere のアップグレード

データベースミラーリングシステムを使用するための SQL Anywhere のアップグレード (EBF の適用を含む) については、「データベースミラーリングシステムでの SQL Anywhere ソフトウェアとデータベースのアップグレード」『SQL Anywhere 12 変更点とアップグレード』を参照してください。

### 参照

- 「LOAD TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「LOAD TABLE 文を使用したデータのインポート」『SQL Anywhere サーバー SQL の使用法』

## データベースミラーリングの利点

ミラーリングには次のような利点があります。

- 監視サーバーが存在する場合、プライマリからミラーへのフェールオーバーは自動で実行される。同期モードで実行している場合、コミットされたトランザクションがフェールオーバー中に失われることはありません。

- ミラーサーバーにトランザクションログが適用済みなので、フェールオーバーが非常に高速になる。ミラーがプライマリの障害を検知すると、コミットされていないトランザクションをすべてロールバックし、データベースを使用可能にします。
- 特別なハードウェア (共有ディスクなど) が不要。
- 特別なソフトウェア (クラスタリング用など) が不要。
- 特定のオペレーティングシステムバージョン要件がない。
- サーバー同士が地理的に近い場所に設置されている必要がない。逆に、互いに距離を置くことで、火事などの災害に対する安全策が強化されます。ただし、サーバー間のネットワーク接続が低速である場合、クライアントアプリケーションのパフォーマンスが影響を受ける可能性があります。
- ミラーリングシステムを構成するデータベースサーバーは、他のデータベースの実行にも使用できる。

## 監視サーバー

監視サーバーは、プライマリサーバー選定におけるサーバー間の競合を解消します。監視サーバーがないと、サーバー A の起動時にサーバー B が使用できなかった場合に、サーバー A は自身のデータベースファイルのコピーが最新の状態なのかを判断できません。最新ではないファイルを使用してデータベースを起動すると、もう一方のデータベースのコピーに適用されてコミットされたトランザクションが失われます。また、2 つのパートナーサーバーが接続を再確立した後では、もう一方のデータベースのコピーをミラーリングに使用できなくなります。

監視サーバーは、起動時の競合を解消するほか、2 つのサーバーが、サーバー間の通信リンクが壊れたままで引き続き実行されている場合についても対応します。監視サーバーがなかった場合、どちらのサーバーもデータベースの所有権を取得しようとします。この場合も、トランザクションが失われ、それぞれ互換性のないデータベースになってしまいます。監視サーバーがあれば、プライマリサーバーはデータベースの所有権が存続することを証明して、クライアントにアクセスを提供し続けることができます。プライマリサーバーがミラーと監視サーバーのどちらとも通信できなくなった場合は、シャットダウンして、どちらかに接続可能になるまで待機する必要があります。

監視サーバーは、複数のミラーリングシステムの監視サーバーとして機能できます。また、他のデータベースのデータベースサーバーとしても機能できます。

## データベースミラーリングモード

データベースミラーリングには次の 3 種類の実行モードがあります。

- 同期
- 非同期
- 非同期フルページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

各モードは、トランザクションがミラーサーバーで記録されるタイミングと方法を制御します。設定には、synchronization_mode オプションの値を設定する SET MIRROR OPTION 文を使用します。デフォルトは同期モードです。

コピーノードへのログページの送信は、選択されたモードに関係なく常に非同期で行われます。

データベースミラーリングシステムで使用する同期実行モードを選択する場合は、フェールオーバーの発生時にリカバリ速度とデータの状態とでどちらを優先するかを決定する必要があります。

データベースミラーリングのモードは、MirrorMode データベースプロパティの値を問い合わせることによって確認できます。

```
SELECT DB_PROPERTY( 'MirrorMode' );
```

### 同期モード

同期モードでは、コミットされたトランザクションはミラーサーバーに記録されることが保証されます。プライマリサーバーで障害が発生した場合、ミラーサーバーが引き継いだときに、コミットされたトランザクションが失われることはありません。このモードでは、プライマリサーバーはトランザクションがコミットされるとトランザクションログページをミラーに送信します。ミラーサーバーは、送信されたページをトランザクションログのコピーに書き込むと、転送の受信確認を行います。プライマリサーバーは、受信確認を受信してからアプリケーションに応答します。

同期モードを使用することで「トランザクションの安全性」が保証されます。これは、パートナーサーバーが同期された状態になり、プライマリは、ミラーに送信した変更の受信確認を待ってから、処理を続行するからです。

### 非同期モード

非同期モードでは、コミットされたトランザクションがミラーサーバーに記録されることは保証されません。このモードでは、プライマリサーバーはトランザクションがコミットされるとトランザクションログページをミラーに送信します。この時点で、プライマリはミラーからの受信確認を待たずに、COMMIT の完了をアプリケーションに応答します。このため、プライマリサーバーで障害が発生した場合、ミラーサーバーが引き継いだときにコミット済みのトランザクションが一部失われる可能性があります。

### 非同期フルページモード

非同期フルページ (または page) モードでは、ページは COMMIT の実行時にではなく、ページが満杯になったときに送信されます。これにより、2 つのデータベースサーバー間のトラフィック量が減り、プライマリサーバーのパフォーマンスが向上します。現在のログページは、pagetimeout パラメーターに指定した秒数を過ぎてもミラーに送信されなかった場合に、ページが満杯でなくても送信されます。デフォルトの値は 5 秒です。このモードを使用すると、プライマリサーバーがダウンし、ミラーサーバーがデータベースの所有権を引き継ぐ場合に、コミットされたトランザクションが失われる可能性がある状態に置かれている時間を制限できます。非同期フルページモードは非同期操作の一種であるため、プライマリサーバーはミラーからの受信確認を待機しません。

非同期モードおよび非同期フルページモードは、同期モードより高速ですが、上記の理由により信頼性は低下します。非同期モードおよび非同期フルページモードでは、プライマリサーバーに適用されたコミット済みトランザクションがミラーサーバーにすべて適用されているとは限らないので、プライマリサーバーからミラーサーバーへのフェールオーバーは自動ではありません。このため、非同期のどちらかのモードを使用する場合は、ミラーサーバーはデフォルトで、プライマリに障害が発生してもデータベースの所有権を取得できません。この場合に (トランザクションが失われる可能性があったとしても) 自動フェールオーバーを使用したい場合は、SET MIRROR OPTION 文を使用して auto_failover オプションを on に設定します。それ以外の場合、障害が発生したサーバーは、再起動後にトランザクションが失われたかどうかを確認します。トランザクションが失われていた場合、データベースサーバーメッセージログにメッセージを書き込んで、データベースをシャットダウンします。現在のデータベースとトランザクションログは、ミラーリングを続行する前にバックアップを使用して置き換える必要があります。

非同期モードまたは非同期フルページモードで障害が発生した後にサーバーを再起動する方法については、「プライマリサーバー障害」1170 ページを参照してください。

**注意**
非同期モードまたは非同期フルページモードを使用している場合は、auto_failover ミラーリングオプションを on に設定することをおすすめします。それによって、プライマリサーバーで障害が発生した場合、ミラーサーバーが自動的にプライマリサーバーとなります。

synchronize_mirror_on_commit データベースオプションでは、非同期モードまたは非同期フルページモードにおいて、データベースの変更がミラーサーバーへ送信されたことがどの時点で保証されるかについて制御できます。このオプションを On に設定すると、COMMIT が発生するたびに、トランザクションログに記録されたすべての変更がミラーサーバーへ送信され、その変更がミラーサーバーによって受信されると、ミラーサーバーからプライマリサーバーへ受信確認が送信されます。このオプションは、SET TEMPORARY OPTION を使用して特定のトランザクションに対して設定できます。また、ログインプロシージャーで APPINFO 文字列を調べ、このオプションを特定のアプリケーションだけに設定すると、効果的な場合もあります。

SQL Anywhere では、データベースミラーリングシステムでフェールオーバーが発生したときに、使用しているモードに関係なく起動するシステムイベントをサポートしています。このようなイベントを使用することで、フェールオーバーの発生時に管理者に通知することなどができます。「データベースミラーリングにおけるシステムイベント」1173 ページを参照してください。

**参照**


- 「synchronize_mirror_on_commit オプション」694 ページ
- 「SET OPTION 文」『SQL Anywhere サーバー SQL リファレンス』
- 「SET MIRROR OPTION 文」『SQL Anywhere サーバー SQL リファレンス』


## 同期ステータス

同期モードを使用しているミラーリングシステムは、「同期中」または「同期」のどちらかのステータスになります。

パートナーサーバーのどれかが起動し、ミラーとして動作することになった場合、まず取得していないログページをプライマリサーバーに要求します。この処理には、プライマリサーバー上で

現在アクティブなログ以外のログファイルからページをコピーすることも含まれます。ミラーは、ページを受信しては、そこに含まれている変更を自身のデータベースのコピーに適用します。ミラーがプライマリからすべてのページを受信すると、プライマリとミラーは同期した状態になります。この時点以降、プライマリ上でコミットされたあらゆる変更はミラーに送信され、ミラーによって受信確認される必要が生じます。

非同期モードと非同期フルページモードでは、ミラーは上記と同様にログページを要求します。ただし、2 つのサーバーが同期した状態になることはありません。ミラーがプライマリから取得できるすべてのログページを要求すると、プライマリは、ページが更新されたらすべてミラーに送信するよう通知されます。

## ステータス情報ファイル

高可用性システム内の各パートナーと監視サーバーのそれぞれがステータス情報ファイルを管理し、ミラーリングシステムのステータスについて各サーバーの視点から記録を残します。

ステータス情報ファイルは、起動時に、サーバーが引き受けるロールの決定に使用されます。サーバーのローカルステータスは、データベースミラーリングシステム内の他のサーバーのステータスと比較されます。

ミラーリングシステムの各サーバーのステータス情報ファイルは、CREATE MIRROR SERVER 文の state_file オプションを使用して必ず指定する必要があります。「CREATE MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

ステータス情報ファイルに含まれている情報は次のとおりです。

| フィールド | 説明 |
|---|---|
| **Owner** | どのデータベースサーバーがプライマリサーバーかを示す。 |
| **State** | サーバーがログページの受信中なのか、または最新状態なのかを示す同期ステータス (「同期中」または「同期」のどちらか) を示す。「同期ステータス」1144 ページを参照してください。 |
| **Mode** | 同期実行モード (同期、非同期、非同期フルページのいずれか) を示す。「データベースミラーリングモード」1142 ページを参照してください。 |

| フィールド | 説明 |
| --- | --- |
| **Sequence** | データベースミラーリングシステムでフェールオーバーが発生した回数を示す。このシーケンス番号はロールの切り替えが発生するたびに 1 ずつ増分されます。これは、サーバーの視点でミラーリングシステムの状態が最新かどうかを判断するのに役立ちます。「データベースミラーリング」1137 ページを参照してください。 |

ステータス情報ファイルの内容の例を次に示します。

```
[demo]
Owner=server2
State=synchronizing
Mode=asynchronous
Sequence=35
```

ステータス情報ファイルがない場合は、自動的に作成されます。ステータス情報ファイルの変更は、データベースサーバーのみが行います。ただし、ミラーリングシステムをリセットしたり、再配備するときには、ステータスファイルを削除します。

# チュートリアル：データベースミラーリングの使用

このチュートリアルでは、データベースミラーリングの設定手順、フェールオーバーへの対処方法、SQL Anywhere モニターからのデータベースミラーリングシステムのモニタリング方法を示します。このチュートリアルでは、すべてのデータベースサーバーが同じコンピューター上で実行されていることを想定しています。ただし、運用環境では、一般に各データベースサーバーは別々のコンピューターにインストールされています。

異なるコンピューターで実行されているデータベースサーバーでこのチュートリアルを使用する場合は、接続文字列内のローカルホストへの参照を、実際のコンピューター名に変更する必要があります。

## レッスン 1：データベースミラーリングシステムの作成

**♦ データベースミラーリングシステムを設定する**

1. ディレクトリ *c:¥server1*、*c:¥server2*、*c:¥arbiter* を作成します。

2. 次のコマンドを実行して、サンプルデータベースのデータが格納された *mirror_demo.db* という名前のデータベースを作成します。

   ```
   newdemo c:¥server1¥mirror_demo.db
   ```

3. 次のコマンドを実行して、第 1 データベースサーバーを起動します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
dbsrv12 -n mirror_server1 -x "tcpip(PORT=6871;DOBROAD=no)" -su sql "c:
¥server1¥mirror_demo.db" -xp on
```

- **-n** データベースサーバーに mirror_server1 という名前を付けます。
- **-x** データベースサーバーの実行で使用するポート番号を指定します。
- **-su** ユーティリティデータベースのパスワードを指定します。-su オプションを使用して、ユーティリティデータベースのパスワードを指定することをおすすめします。これにより、ユーティリティデータベースに接続してデータベースサーバーをシャットダウンしたり、必要に応じてミラーサーバーを強制的にプライマリサーバーにしたりできるようになります。「ミラーリングシステムのデータベースサーバーの停止」1168 ページと「データベースサーバーの強制プライマリサーバー化」1166 ページを参照してください。
- **-xp on** データベースサーバーがデータベースミラーリングシステムに参加可能であることを示します。

4. Interactive SQL からデータベースに接続します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=mirror_server1"
```

5. 次の文を実行して、プライマリサーバーとミラーサーバーを定義します。

```
CREATE MIRROR SERVER mirror_demo_primary
AS PRIMARY
connection_string='SERVER=mirror_demo_primary;HOST=localhost:6871,localhost:6872';

CREATE MIRROR SERVER mirror_demo_mirror
AS MIRROR
connection_string='SERVER=mirror_demo_mirror;HOST=localhost:6871,localhost:6872';
```

6. 次の文を実行して、mirror_server1 と mirror_server2 をミラーリングシステムにおけるパートナーとして定義します。

```
CREATE MIRROR SERVER mirror_server1
AS PARTNER
connection_string='SERVER=mirror_server1;host=localhost:6871'
state_file='c:¥¥server1¥¥server1.state';

CREATE MIRROR SERVER mirror_server2
AS PARTNER
connection_string='SERVER=mirror_server2;host=localhost:6872'
state_file='c:¥¥server2¥¥server2.state';
```

7. 次の文を実行して、監視サーバーを定義します。

```
CREATE MIRROR SERVER demo_arbiter
AS ARBITER
connection_string ='SERVER=demo_arbiter;HOST=localhost:6870';
```

8. 次の文を実行して、データベースの認証文字列を設定します。

```
SET MIRROR OPTION authentication_string='abc';
```

9. Interactive SQL との接続を切断します。

10. 次のコマンドを実行して、*c:¥server1* にあるデータベースファイルとトランザクションログのコピーを作成し、*c:¥server2* に追加します。

```
dbbackup -c "UID=DBA;PWD=sql;SERVER=mirror_server1;DBN=mirror_demo" c:¥server2
```

11. 次のコマンドを実行して、第 2 データベースサーバーを起動します。

```
dbsrv12 -n mirror_server2 -x "tcpip(PORT=6872;DOBROAD=no)" -su sql "c:
¥server2¥mirror_demo.db" -xp on
```

このコマンドラインは、以下に示す dbsrv12 のオプションを指定します。

- **-n** データベースサーバーに mirror_server2 という名前を付けます。
- **-x** データベースサーバーの実行で使用するポート番号を指定します。
- **-su** ユーティリティデータベースのパスワードを指定します。-su オプションを使用して、ユーティリティデータベースのパスワードを指定することをおすすめします。これにより、ユーティリティデータベースに接続してデータベースサーバーをシャットダウンしたり、必要に応じてミラーサーバーを強制的にプライマリサーバーにしたりできるようになります。「ミラーリングシステムのデータベースサーバーの停止」1168 ページと「データベースサーバーの強制プライマリサーバー化」1166 ページを参照してください。
- **-xp on** データベースサーバーがデータベースミラーリングシステムに参加可能であることを示します。

12. 次のコマンドを実行して、監視データベースサーバーを起動します。

```
dbsrv12 -n demo_arbiter -su sql -x "tcpip(PORT=6870;DOBROAD=no)" -xf "c:¥arbiter¥arbiter.state"
-xa "AUTH=abc;DBN=mirror_demo"
```

- **-n** データベースサーバーに demo_arbiter という名前を付けます。
- **-su** ユーティリティデータベースのパスワードを指定します。-su オプションを使用して、ユーティリティデータベースのパスワードを指定することをおすすめします。これにより、ユーティリティデータベースに接続してデータベースサーバーをシャットダウンしたり、必要に応じてミラーサーバーを強制的にプライマリサーバーにしたりできるようになります。「ミラーリングシステムのデータベースサーバーの停止」1168 ページと「データベースサーバーの強制プライマリサーバー化」1166 ページを参照してください。
- **-x** データベースサーバーに対し、TCP/IP 通信にポート 6870 を使用するよう指示します。他のサーバーも TCP/IP を使用しますが、通信には別のポートを使用します。
- **-xf** arbiter のステータス情報ファイルのロケーションを指定します。
- **-xa** ミラーリング対象のデータベースの名前と、監視サーバーに対する認証文字列 (ここでは abc) を指定します。この認証文字列を、データベースミラーリングシステムのすべてのサーバー (監視サーバー、プライマリサーバー、ミラーサーバー) で使用する必要があります。

13. 次のコマンドを実行して、Interactive SQL を起動し、プライマリサーバー用の代替サーバー名を使用してプライマリサーバーに接続します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=mirror_demo_primary;HOST=localhost:6871,localhost:
6872"
```

14. 次の文を実行して、接続したデータベースサーバーを確認します。

```
SELECT PROPERTY( 'ServerName' );
```

プライマリサーバーとして機能しているサーバーの名前が表示されます。

例：mirror_server1

15. 次の文を実行して、データをデータベースに追加します。

```
CREATE TABLE test (col1 INTEGER, col2 CHAR(32));
INSERT INTO test VALUES(1, 'Hello from server1');
COMMIT;
```

16. 「レッスン 2：データベースミラーリングシステムでのフェールオーバーの開始」1149 ページに進みます。

**参照**


- 「データベースミラーリング」1137 ページ
- 「データベースミラーリングシステムの設定」1160 ページ
- 「チュートリアル：監視サーバーを共有する複数のデータベースでデータベースミラーリングを使用する」1154 ページ


## レッスン 2：データベースミラーリングシステムでのフェールオーバーの開始

**♦ データベースミラーリングシステムでフェールオーバーを作成する**

1. フェールオーバーを開始します。これを行うには、次のいずれかの方法でプライマリサーバー (現在は mirror_server1) を停止します。

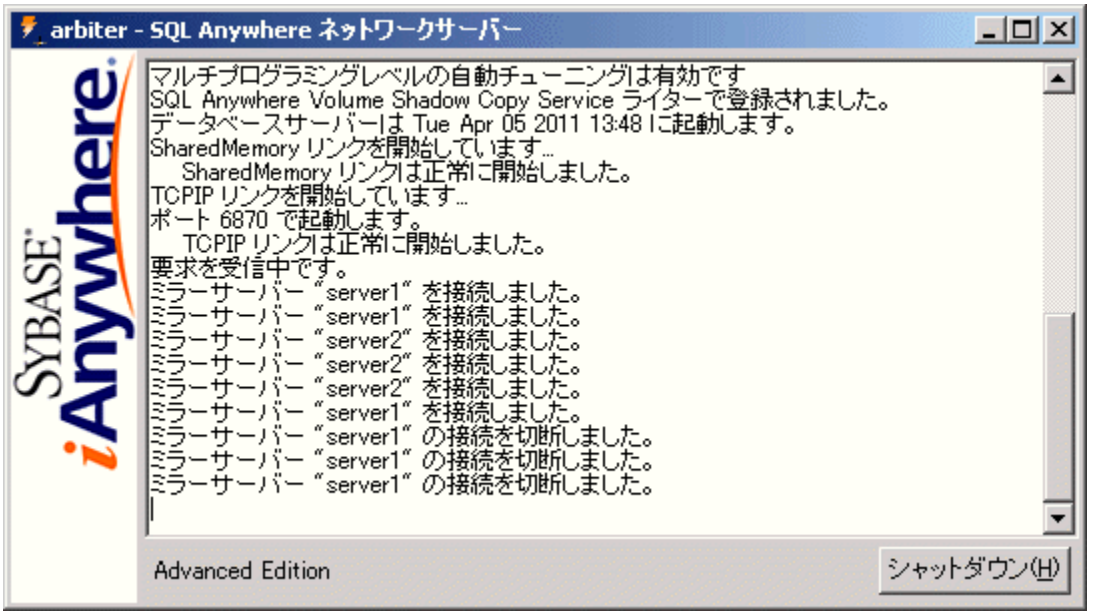
   - データベースサーバーメッセージウィンドウで **[シャットダウン]** をクリックする。
   - Windows のタスク マネージャーを使用してタスクを終了します。
   - 次のコマンドを実行します。

     ```
     dbstop -y -c "UID=DBA;PWD=sql;SERVER=mirror_demo_primary"
     ```

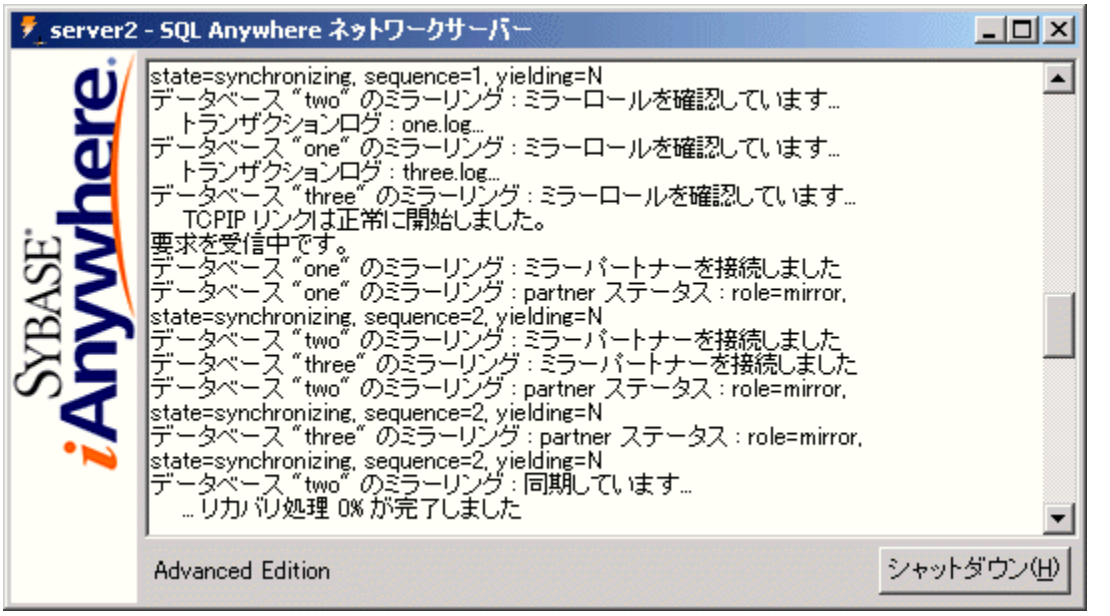
   「ミラーリングシステムのデータベースサーバーの停止」1168 ページを参照してください。

   データベースサーバーの 1 つの接続が有効であるという警告メッセージが表示された場合は、**[はい]** をクリックしてシャットダウンします。

   arbiter のデータベースサーバーメッセージウィンドウに、プライマリサーバーが切断されたことを示すメッセージが表示されます。




mirror_server2 のデータベースサーバーメッセージウィンドウには、mirror_server2 が新しいプライマリサーバーであることを示すメッセージが表示されます。




2. Interactive SQL を閉じます。エラーメッセージを受信した場合は **[OK]** をクリックします。

3. 次のコマンドを実行して、Interactive SQL を再起動します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=mirror_demo_primary;HOST=localhost:6871,localhost:
6872"
```

4. 次の文を実行して、これまでミラーサーバーとして機能していたサーバー (mirror_server2) が、現在はプライマリサーバーとして機能していることを確認します。

```
SELECT PROPERTY ( 'ServerName' );
```

5. 次の文を実行して、すべてのトランザクションがミラーサーバーにミラーリングされたかどうかを確認します。

```
SELECT * FROM test;
```

6. Interactive SQL との接続を切断します。

7. 「レッスン 3：データベースミラーリングシステムのモニタリング」1151 ページに進みます。

**参照**

- 「データベースミラーリングの切断された接続のシナリオ」1178 ページ

## レッスン 3：データベースミラーリングシステムのモニタリング

**♦ データベースミラーリングシステムリソースをモニターに追加する**

1. SQL Anywhere モニターを起動します。
   - **モニター Developer Edition の起動 (Windows の場合)**　**[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[SQL Anywhere モニター]** をクリックします。
   - **モニター Developer Edition の起動 (Linux の場合)**　モニターのインストールディレクトリの *bin32* または *bin64* ディレクトリから *samonitor.sh* スクリプトを実行します。

     ```
     samonitor.sh launch
     ```

2. デフォルトの管理者ユーザーとしてモニターにログインします。
   - **[ユーザー名]** フィールドに **admin** と入力し、**[パスワード]** フィールドに **admin** と入力します。

3. データベースミラーリングシステムをモニターの **[リソースリスト]** に追加します。
   a. 左のサイドバーにある **[ツール]** » **[管理]** をクリックします。
   b. **[リソース]** をクリックして、**[追加]** をクリックします。
   c. **[SQL Anywhere サーバー]** **[次へ]** をクリックします。
   d. **[名前]** フィールドに **mirrorsystemdemo** と入力し、**[次へ]** をクリックします。
   e. **[ホスト]** フィールドでプライマリサーバとミラーサーバーのホスト名とポート番号をカンマ区切りのリストで指定します。つまり、**localhost:6871, localhost:6872** と入力します。
   f. **[サーバー]** フィールドにプライマリサーバーの代替サーバー名を入力します。つまり、**mirror_demo_primary** と入力します。
   g. **[作成]** をクリックします。

4. 必要な認証情報が要求されたら、**[DBA ユーザー ID]** フィールドに **DBA** と入力し、**[パスワード]** フィールドに **sql** と入力します。**[OK]** をクリックします。

データベースをモニタリング対象のリソースとして追加する場合は、データベースの DBA ユーザー ID とパスワードを指定する必要があります。これらのクレデンシャルは次の操作を行う場合に使用されます。

- データベースに接続します。
- **sa_monitor_user** という名前の新しいユーザーを作成します。モニターは、sa_monitor_user を使用してデータベースに接続し、モニタリングします。
- データベースをモニタリングするために **sa_monitor_user** が必要とするデータベースオブジェクトをインストールします。インストールされるオブジェクトのリストについては、「モニターがインストールしたデータベースオブジェクトのリスト」1311 ページを参照してください。
- モニターから DBA クレデンシャルを破棄します。**sa_monitor_user** はモニタリングされるデータベースに追加されるので、そのデータベース外部のどこかに DBA クレデンシャルを格納する必要はありません。「モニターがインストールしたデータベースオブジェクトのリスト」1311 ページを参照してください。

リソースが追加され、モニタリングが自動的に開始されます。

5. **[閉じる]** をクリックします。
6. **[閉じる]** をクリックします。
7. **[ダッシュボード]** » **[概要]** をクリックします。

   **[リソースリスト]** に mirrorsystemdemo リソースが表示されます。
8. **[mirrorsystemdemo]** をクリックすると、mirrorsystemdemo ダッシュボードが開き、収集されたメトリックが表示されます。

**♦ [SQL Anywhere スケールアウトトポロジ] ウィジェットの追加**

1. mirrorsystemdemo ダッシュボードの右上端で、**[カスタマイズ]** » **[ウィジェットの追加]** » **[SQL Anywhere スケールアウトトポロジ]** をクリックします。
2. **[次へ]** をクリックします。
3. **[このウィジェットの名前を指定してください。]** フィールドに **mirrorsystemwidget** と入力します。
4. **[どのようなリソースに関心がありますか ?]** フィールドで **[mirrorsystemdemo]** をクリックします。
5. **[作成]** をクリックします。

ダッシュボードに新しいウィジェットが表示されます。

| サーバー名 | タイプ | 状態 | 接続 | 最終更新 |
|---|---|---|---|---|
| ▼ scaleout_root_demo | partner - primary | connected | 1 | 2011/04/05 14:22 |
| scaleout_child_demo | copy | connected | 1 | 2011/04/05 14:22 |

トポロジウィジェットでは、次のカラムとその使用できる値が表示されます。

- **[サーバー名]**　ミラーリングシステムにデータベースサーバーの名前が表示されます。
- **[タイプ]**　サーバーのロールが一覧表示され、次の値の 1 つが表示されます。

  **partner - primary**　この値によって、現在プライマリとして動作しているサーバーが識別されます。

  **partner - mirror**　この値によって、現在ミラーとして動作しているサーバーが識別されます。

  **[arbiter]**　この値によって、監視サーバーが識別されます。
- **[状態]**　各サーバーの接続状況が表示されます。
- **[接続]**　サーバーへの現在の接続数が表示されます。
- **[最終更新]**　サーバーのステータスが更新された最新の時刻が表示されます。

「レッスン 4：モニターからのログアウトと、データベースミラーリングシステムのシャットダウン」1153 ページに進みます。

**参照**


- 「SQL Anywhere モニター」1217 ページ
- 「[概要] ダッシュボード」1240 ページ
- 「[SQL Anywhere スケールアウトトポロジ] ウィジェット」1245 ページ
- 「ウィジェット」1244 ページ


## レッスン 4：モニターからのログアウトと、データベースミラーリングシステムのシャットダウン

**♦ モニターをログアウトして、データベースミラーリングシステムをシャットダウンする**

1. **[ログアウト]** をクリックしてモニターからログアウトします。
2. SQL Anywhere モニターを表示していたブラウザーウィンドウを閉じます。
3. モニターを終了し、モニタリングを停止します。

- Windows では、システムトレイで [SQL Anywhere モニター] アイコンを右クリックし、**[SQL Anywhere モニターの終了]** をクリックします。
- Linux では、モニターのインストールディレクトリの *bin32* または *bin64* ディレクトリから *samonitor.sh* スクリプトを実行します。

```
samonitor.sh stop
```

モニターは、すべてのリソースについてメトリックの収集を停止します。

4. 次のいずれか 1 つの操作を実行することで、SQL Anywhere データベースミラーリングシステムをシャットダウンします。

- demo_arbiter、mirror_server1、mirror_server2 データベースサーバーのデータベースサーバーメッセージウィンドウで、**[シャットダウン]** をクリックします。
- 次のコマンドを実行します。

```
dbstop -y -c "UID=DBA;PWD=sql;SERVER=mirror_demo_primary"
dbstop -y -c "UID=DBA;PWD=sql;SERVER=mirror_demo_mirror"
dbstop -y -c "UID=DBA;PWD=sql;SERVER=demo_arbiter;DBN=utility_db"
```

- (任意) *c:¥server1*、*c:¥server2*、*c:¥arbiter* の各ディレクトリを削除します。

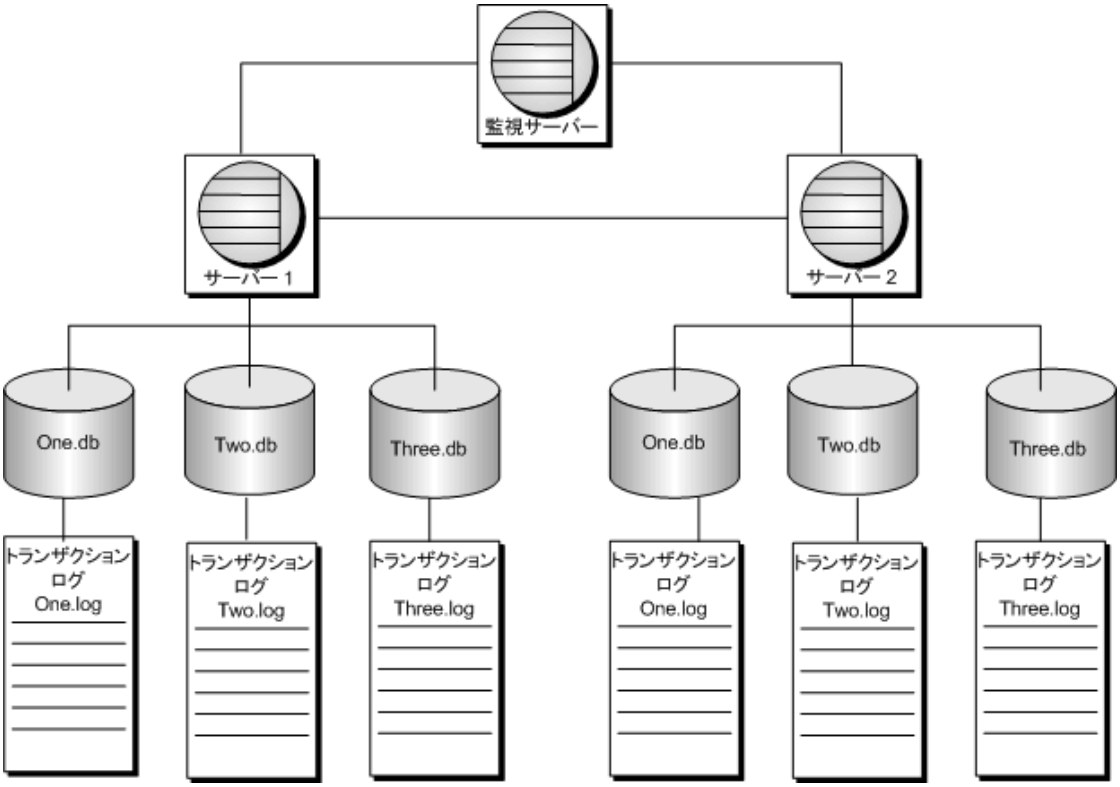
## チュートリアル：監視サーバーを共有する複数のデータベースでデータベースミラーリングを使用する

**注意**
異なるコンピューターで実行されているデータベースサーバーでこのチュートリアルを使用する場合は、接続文字列内のローカルホストへの参照を、実際のコンピューター名に変更する必要があります。

*%SQLANYSAMP12%¥SQLAnywhere¥DBMirror* に、スケールアウトシステムでデータベースミラーリングを使用するサンプルがあります。このサンプルは、1 台のコンピューターまたは複数のコンピューターで実行できます。

この構成では、プライマリサーバーとミラーサーバーのそれぞれが、ミラーリングシステムに参加している 3 つの個別データベースを実行します。3 つのミラーリングシステムはすべて同じ監視サーバーと通信します。各ミラーリングシステムでは、CREATE MIRROR SERVER 文の PRIMARY 句と MIRROR 句で指定された一意の代替サーバー名が使用されます。このような構成では、プライマリサーバー、ミラーサーバー、監視サーバーを、それぞれ別のコンピューターで実行することができます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1




プライマリサーバーが使用できなくなった場合は、ロールの切り替えが発生し、ミラーサーバーがデータベースの所有権を取得します。そのミラーサーバーがプライマリサーバーになります。クライアントはプライマリサーバーへの接続を再確立する必要があります。プライマリサーバーへの接続を再確立するために指定する必要があるのは、代替サーバー名だけです。また、この構成には、1 つのデータベースに発生した障害からの保護機能もあります。プライマリサーバーで実行されているデータベースが使用できなくなると、ロールの切り替えが行われ、障害が発生したデータベースの所有権をミラーサーバーが取得します。このデータベースに対してのみ、ミラーサーバーがプライマリサーバーになります。クライアントは代替サーバー名を使用して、このデータベースのプライマリサーバーに対する接続を再確立する必要があります。

**♦ 3 つのデータベースと 1 つの監視サーバーで構成されるミラーリングシステムの設定**

1. 次のディレクトリを作成します。
   - *c:¥server1*
   - *c:¥server2*
   - *c:¥arbiter*
2. 次のコマンドを実行します。

   ```
   newdemo c:¥server1¥one.db
   ```

3. 次のコマンドを実行します。

   ```
   newdemo c:¥server1¥two.db
   ```

4. *c:¥server1* ディレクトリから、次のコマンドを実行します。

```
newdemo c:¥server1¥three.db
```

5. server1 という名前のデータベースサーバーを起動します。

```
dbsrv12 -n server1 -x tcpip(PORT=6871) -su sql c:¥server1¥one.db -xp on c:¥server1¥two.db -xp
on c:¥server1¥three.db -xp on
```

-su オプションを使用して、ユーティリティデータベースのパスワードを指定することをおすすめします。これにより、ユーティリティデータベースを使用してデータベースサーバーをシャットダウンしたり、必要に応じてミラーサーバーを強制的にプライマリサーバーにしたりできるようになります。「ミラーリングシステムのデータベースサーバーの停止」1168 ページと「データベースサーバーの強制プライマリサーバー化」1166 ページを参照してください。

6. Interactive SQL からデータベース one に接続し、必要なミラーリングオブジェクトを定義します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=server1;DBN=one"
```

a. server1 をデータベース one のパートナーサーバーとして定義します。

```
CREATE MIRROR SERVER server1
AS PARTNER
connection_string='SERVER=server1;host=localhost:6871'
state_file='c:¥¥server1¥¥server1state.txt';
```

b. データベースサーバー primary_one をデータベース one の論理プライマリサーバーとして定義します。

```
CREATE MIRROR SERVER primary_one
AS PRIMARY
connection_string='SERVER=primary_one;host=localhost:6871,localhost:6872';
```

c. server2 をデータベース one のパートナーサーバーとして定義します。

```
CREATE MIRROR SERVER server2
AS PARTNER
connection_string='SERVER=server2;host=localhost:6872'
state_file='c:¥¥server2¥¥server2state.txt';
```

d. データベースサーバー mirror_one をデータベース one の論理ミラーサーバーとして定義します。

```
CREATE MIRROR SERVER mirror_one
AS MIRROR
connection_string='SERVER=mirror_one;host=localhost:6871,localhost:6872';
```

e. 監視サーバーを定義します。

```
CREATE MIRROR SERVER arbiter
AS ARBITER
connection_string='SERVER=arbiter;HOST=localhost:6870';
```

f. データベースミラーリングシステムのミラーリングオプションを設定します。

```
SET MIRROR OPTION authentication_string='abc';
```

g. Interactive SQL との接続を切断します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

h. *c:¥server2* ディレクトリにデータベースのバックアップコピーを作成します。

```
dbbackup -c "UID=DBA;PWD=sql;SERVER=server1;DBN=one" c:¥server2
```

7. Interactive SQL からデータベース two に接続し、必要なミラーリングオブジェクトを定義します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=server1;DBN=two"
```

a. server1 をデータベース two のパートナーサーバーとして定義します。

```
CREATE MIRROR SERVER server1
AS PARTNER
connection_string='SERVER=server1;host=localhost:6871'
state_file='c:¥¥server1¥¥server1state.txt';
```

b. データベースサーバー primary_two をデータベース two の論理プライマリサーバーとして定義します。

```
CREATE MIRROR SERVER primary_two
AS PRIMARY
connection_string='SERVER=primary_two;host=localhost:6871,localhost:6872';
```

c. server2 をデータベース two のパートナーサーバーとして定義します。

```
CREATE MIRROR SERVER server2
AS PARTNER
connection_string='SERVER=server2;host=localhost:6872'
state_file='c:¥¥server2¥¥server2state.txt';
```

d. データベースサーバー mirror_two をデータベース two の論理ミラーサーバーとして定義します。

```
CREATE MIRROR SERVER mirror_two
AS MIRROR
connection_string='SERVER=mirror_two;host=localhost:6871,localhost:6872';
```

e. 監視サーバーを定義します。

```
CREATE MIRROR SERVER arbiter
AS ARBITER
connection_string='SERVER=arbiter;HOST=localhost:6870';
```

f. データベースミラーリングシステムのミラーリングオプションを設定します。

```
SET MIRROR OPTION authentication_string='def';
```

g. Interactive SQL との接続を切断します。

h. *c:¥server2* ディレクトリにデータベースのバックアップコピーを作成します。

```
dbbackup -c "UID=DBA;PWD=sql;SERVER=server1;DBN=two" c:¥server2
```

8. Interactive SQL からデータベース three に接続し、必要なミラーリングオブジェクトを定義します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=server1;DBN=three"
```

a. server1 をデータベース three のパートナーサーバーとして定義します。

```
CREATE MIRROR SERVER server1
AS PARTNER
```

```
connection_string='SERVER=server1;host=localhost:6871'
state_file='c:¥¥server1¥¥server1state.txt';
```

b. データベースサーバー primary_three をデータベース three の論理プライマリサーバーとして定義します。

```
CREATE MIRROR SERVER primary_three
AS PRIMARY
connection_string='SERVER=primary_three;host=localhost:6871,localhost:6872';
```

c. server2 をデータベース three のパートナーサーバーとして定義します。

```
CREATE MIRROR SERVER server2
AS PARTNER
connection_string='SERVER=server2;host=localhost:6872'
state_file='c:¥¥server2¥¥server2state.txt';
```

d. データベースサーバー mirror_three をデータベース three の論理ミラーサーバーとして定義します。

```
CREATE MIRROR SERVER mirror_three
AS MIRROR
connection_string='SERVER=mirror_three;host=localhost:6871,localhost:6872';
```

e. 監視サーバーを定義します。

```
CREATE MIRROR SERVER arbiter
AS ARBITER
connection_string='SERVER=arbiter;HOST=localhost:6870';
```

f. データベースミラーリングシステムのミラーリングオプションを設定します。

```
SET MIRROR OPTION authentication_string='ghi';
```

g. Interactive SQL との接続を切断します。

h. *c:¥server2* ディレクトリにデータベースのバックアップコピーを作成します。

```
dbbackup -c "UID=DBA;PWD=sql;SERVER=server1;DBN=three" c:¥server2
```

9. server2 という名前のデータベースサーバーを起動します。

```
dbsrv12 -n server2 -x tcpip(PORT=6872) -su sql c:¥server2¥one.db -xp on c:¥server2¥two.db -xp
on c:¥server2¥three.db -xp on
```

10. 監視サーバーを起動します。

```
dbsrv12 -n arbiter -su sql -x tcpip(port=6870) -xf c:¥arbiter¥arbiterstate.txt -xa
"AUTH=abc,def,ghi;DBN=one,two,three"
```

server2 を起動した後、server1 のデータベースサーバーメッセージウィンドウに、server1 がデータベース one、two、three のミラーリングシステムにおけるプライマリサーバーであることが表示されます。また、メッセージには、one、two、three のミラーデータベース (パートナー) が server1 に接続されていることも表示されます。

arbiter のメッセージは、server1 と server2 の両方が接続されていることを示します。

11. 次のコマンドを実行して、Interactive SQL を起動し、プライマリサーバーにあるデータベース one に接続します。

```
dbisql -c "UID=DBA;PWD=sql;Server=primary_one;LINKS=TCPIP"
```

12. 次の文を実行して、サンプルデータを SQL Anywhere サンプルデータベースに追加します。

```
CREATE TABLE test (col1 INTEGER, col2 CHAR(32));
INSERT INTO test VALUES(1, 'Hello from server1');
COMMIT;
```

13. 次の文を実行して、接続したデータベースサーバーを確認します。

```
SELECT PROPERTY( 'ServerName' );
```

プライマリサーバーの名前が出力されます。

14. Interactive SQL との接続を切断します。

15. フェールオーバーを開始します。これを行うには、次のいずれかの方法でプライマリサーバーを停止します。

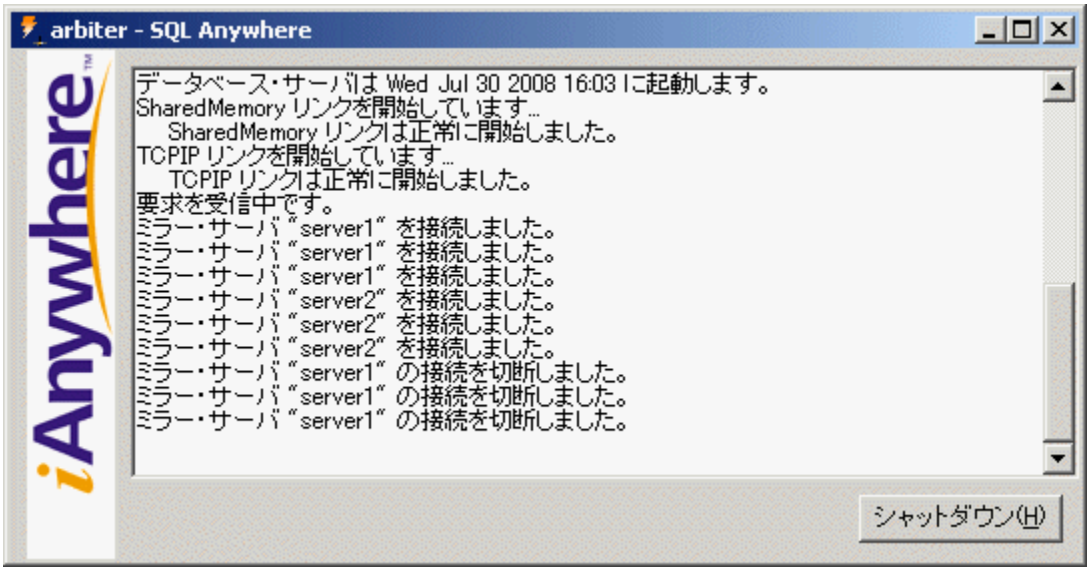
- データベースサーバーメッセージウィンドウで **[シャットダウン]** をクリックします。
- Windows のタスク マネージャーを使用してタスクを終了します。
- 次のコマンドを実行します。

```
dbstop -y -c "UID=DBA;PWD=sql;Server=server1"
```

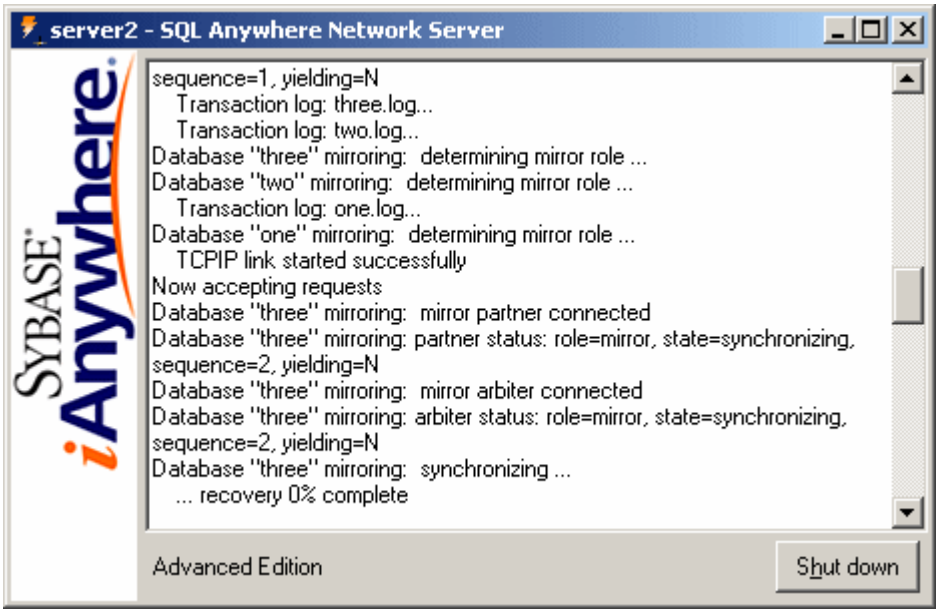
「ミラーリングシステムのデータベースサーバーの停止」1168 ページを参照してください。

データベースサーバーの 1 つの接続が有効であるという警告メッセージが表示された場合は、**[はい]** をクリックしてシャットダウンします。

arbiter のデータベースサーバーメッセージウィンドウに、プライマリサーバーが切断されたことを示すメッセージが表示されます。



server2 のデータベースサーバーメッセージウィンドウには、server2 が新しいプライマリサーバーであることを示すメッセージが表示されます。




16. 次のコマンドを実行して、Interactive SQL を再起動します。

    ```
    dbisql -c "UID=DBA;PWD=sql;Server=primary_one;LINKS=tcpip"
    ```

17. 次の文を実行して、server2 に接続されているかどうかを確認します。

    ```
    SELECT PROPERTY ( 'ServerName' );
    ```

18. 次の文を実行して、すべてのトランザクションがミラーデータベースにミラーリングされたかどうかを確認します。

    ```
    SELECT * FROM test;
    ```

19. Interactive SQL から切断し、arbiter と server2 のデータベースサーバーメッセージウィンドウで **[シャットダウン]** をクリックします。

20. (任意) *c:¥server1*、*c:¥server2*、*c:¥arbiter* の各ディレクトリを削除します。

## データベースミラーリングシステムの設定

ミラーリングシステムに参加するデータベースサーバーを起動する場合は、**-xp on** を指定します。**-su** オプションも使用して、ユーティリティデータベースのパスワードを指定することをおすすめします。これにより、ユーティリティデータベースを使用してデータベースサーバーをシャットダウンしたり、必要に応じてミラーサーバーを強制的にプライマリサーバーにしたりできるようになります。「ミラーリングシステムのデータベースサーバーの停止」1168 ページと「データベースサーバーの強制プライマリサーバー化」1166 ページを参照してください。

SQL Anywhere のアップグレード、またはデータベースのミラーリングに使用しているデータベースの再構築については、「データベースミラーリングシステムでの SQL Anywhere ソフトウェアとデータベースのアップグレード」『SQL Anywhere 12 変更点とアップグレード』を参照してください。

**注意**
データベースミラーリングシステムを設定する場合は、次のすべての例で、localhost とポート番号を対応するデータベースサーバーが実行されるコンピューター名とポートに変更する必要があります。

### ♦ ミラーリングシステムの設定

1. **-su** オプションと **-xp on** オプションを使用して、データベースサーバーでミラーリングするデータベースを起動します。次に例を示します。

```
dbsrv12 -n mirror_server1 -x tcpip(PORT=6871;DOBROAD=no) -su sql
"c:¥server1¥mymirrordb.db" -xp on
```

「-su dbeng12/dbsrv12 サーバーオプション」276 ページと「-xp dbsrv12 データベースオプション」318 ページを参照してください。

2. CREATE MIRROR SERVER 文を使用して、データベースのパートナーサーバーと監視サーバーを定義します。プライマリサーバーとして 1 台のデータベースサーバー、ミラーサーバーとして 1 台のデータベースサーバー、さらに 1 台の監視サーバーを定義します。プライマリサーバーとミラーサーバーは、データベースシステムのパートナーとしても定義します。

   最初の文は、データベースミラーリングシステムのプライマリサーバーとして動作するデータベースサーバーに接続するためにクライアントで使用する名前を定義します。2 つ目の文は、mirror_server1 をデータベースミラーリングシステムのパートナーサーバーとして定義します。

```
CREATE MIRROR SERVER myprimary
AS PRIMARY
connection_string='SERVER=myprimary;HOST=localhost:6871,localhost:6872';

CREATE MIRROR SERVER mirror_server1
AS PARTNER
connection_string='SERVER=mirror_server1;host=localhost:6871'
state_file='c:¥¥server1¥¥server1.state';
```

   次の SQL 文は、データベースサーバーでミラーサーバーとして動作するデータベースサーバーの名前を定義し、データベースミラーリングシステムの 2 つ目のパートナーサーバーも定義します。

```
CREATE MIRROR SERVER mymirror
AS MIRROR
connection_string='SERVER=mymirror;HOST=localhost:6871,localhost:6872';

CREATE MIRROR SERVER mirror_server2
AS PARTNER
connection_string='SERVER=mirror_server2;host=localhost:6872'
state_file='c:¥¥server2¥¥server2.state';
```

> **注意**
> システムのデータベースサーバーを設定するには、プライマリとミラーのロールが必要です。これらのサーバーに付けた名前は、クライアントからデータベースサーバーに接続するときに代替サーバー名として使用されます。どちらか一方のパートナーサーバーは、プライマリサーバーまたはミラーサーバーとして動作できます。

次の SQL 文は、データベースミラーリングシステムの監視サーバーを定義します。

```
CREATE MIRROR SERVER myarbiter
AS ARBITER
connection_string ='SERVER=myarbiter;HOST=localhost:6870';
```

「CREATE MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

3. ミラーリングシステムのミラーリングオプションを設定します。認証文字列の指定は必須です。次に例を示します。

```
SET MIRROR OPTION authentication_string='abc';
```

「SET MIRROR OPTION 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

4. 2 つ目のデータベースサーバーが実行されるコンピューター上のデータベース、現在のトランザクションログ、および他のトランザクションログのコピーを作成します。

   プライマリサーバーコンピューターとミラーサーバーコンピューター上のトランザクションログファイルは、現在のトランザクションログファイルの開始オフセットを含め、同一である必要があります。

   ミラーリングするデータベースが実行されているデータベースサーバーを停止した場合は、ファイルのコピーを作成できます。そうでない場合は、BACKUP DATABASE 文またはバックアップユーティリティ (dbbackup) を使用します。「BACKUP 文」『SQL Anywhere サーバー SQL リファレンス』と「バックアップユーティリティ (dbbackup)」913 ページを参照してください。

5. データベースミラーリングシステムの 2 つ目のデータベースサーバーを起動します。

```
dbsrv12 -n mirror_server2 -x tcpip(PORT=6872;DOBROAD=no) -su sql
"c:¥server2¥mymirrordb.db" -xp on
```

6. 監視サーバーを起動します。

```
dbsrv12 -n myarbiter -su sql
-x "TCPIP(PORT=6870;DOBROAD=no)" -xf "c:¥arbiter¥arbiter.state"
-xa "AUTH=abc;DBN=mymirrordb"
```

これで、クライアントはミラーリングされたデータベースに接続できるようになりました。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**ヒント**
システムを SQL Anywhere モニターのリソースリストに追加することによって、データベースミラーリングシステムをモニターできます。 「レッスン 3：データベースミラーリングシステムのモニタリング」1151 ページを参照してください。

Sybase Central からプライマリデータベースに接続して、データベースミラーリングシステムのデータベースサーバーのステータスを確認できます。**[正常性と統計情報]** ウィンドウ枠にデータベースミラーリング情報が表示されます。「データベースの正常性と統計情報」837 ページを参照してください。

### ミラーサーバー間の接続文字列の設定

ミラーサーバーを作成するときに、ミラーリングシステムのデータベースサーバー間の接続文字列を設定し、DoBroadcast=none を指定することによって、ブロードキャストが実行されないようにすることができます。これによって、データベースサーバーは別のデータベースサーバーが使用できなくなったことを迅速に判断できるようになります。接続文字列に接続先データベースサーバーのポートだけでなく IP アドレスまたはコンピューター名も指定されている場合は、ブロードキャストを実行することなく TCP/IP 接続を確立できます。サーバーが使用できない場合は、接続が失敗し、タイムアウトの必要もなくなります。Host 接続パラメーターを使用した場合も、接続のタイムアウトを指定する必要はありません。次に例を示します。

```
CREATE MIRROR SERVER mirror_one
AS PRIMARY
connection_string='SERVER=mirror_one;host=localhost:6871,localhost:6872';
```

**参照**

- 「Host 接続パラメーター」346 ページ
- 「DoBroadcast (DOBROAD) プロトコルオプション」379 ページ

### ミラーリングされたデータベースサーバーへの接続

ミラーリングされたデータベースに接続する場合、クライアントは、データベースミラーリングシステムのプライマリデータベースサーバーに指定されたサーバー名を使用する必要があります。たとえば、プライマリデータベースサーバーに指定された名前が myprimary の場合、クライアントは接続文字列で接続パラメーター Server=myprimary を指定します。

```
...UID=user12;PWD=x92H4pY;Server=myprimary;HOST=localhost:6871,localhost:6872...
```

プライマリサーバーとミラーサーバーが異なるサブネット上で実行されている場合は、クライアントがプライマリサーバーに接続するために使用する IP アドレスのリストを指定する必要があります。次に例を示します。

```
...UID=user12;PWD=x92H4pY;Server=myprimary;LINKS=tcpip(HOST=ip1,ip2...)...
```

クライアントがプライマリサーバーへの接続を試行し続ける時間を制御するために、RetryConnectionTimeout 接続パラメーターを指定することもできます。「RetryConnectionTimeout (RetryConnTO) 接続パラメーター」364 ページを参照してください。

クライアントの接続先となるサーバーが見つからない場合は、以下を試してください。

1. プライマリサーバーとミラーサーバーを実行しているコンピューターのホスト名を指定します。たとえば、MirrorServ1 および MirrorServ2 という名前のコンピューター上で実行されている場合、クライアントの接続文字列で HOST=MirrorServ1,MirrorServ2 と指定できます。

2. LDAP にサーバーを登録します。「LDAP サーバーを使用した接続」103 ページを参照してください。

3. SQL Anywhere Broadcast Repeater ユーティリティ (dbns12) を使用してサーバーを探します。このユーティリティは、あるサブネット上でのブロードキャストと応答を受信し、それを別のサブネットに再ブロードキャストします。「Broadcast Repeater ユーティリティ (dbns12)」921 ページを参照してください。

## 初期プライマリサーバー

データベースミラーリングシステムを初めて設定するときに、ステータス情報ファイルがなく、データベースとトランザクションログのコピーが同じである場合、どちらのサーバーでもプライマリとして使用できます。この場合は、サーバー名が比較され、名前の小さい方がプライマリサーバーになります。たとえば、server1 は server2 より小さいと見なされます。

最初に起動するときは、ロールに合意できるように、両方のサーバーが稼働し接続されている必要があります。ステータス情報ファイルに記録されている、以前のステータス情報がないため、監視サーバーの存在だけでは不十分です。

通常の起動では、次の入力によって、どのサーバーがプライマリサーバーになるかが決まります。

- ステータス情報ファイルの内容
- 各データベースサーバーのトランザクションログの位置
- 優先プライマリサーバーの指定

**-xp on** を指定して起動したデータベースサーバー (たとえば S1) で、ミラーリングが定義されていないデータベースを起動してからミラーリングを定義した場合、S1 がデータベースの初期プライマリサーバーになります。

**参照**

- 「ステータス情報ファイル」1145 ページ

## 優先データベースサーバー

データベースミラーリングシステムでは、2 つのパートナーサーバーのうちいずれかを優先サーバーとして指定できます。すべてのデータベースサーバーが実行されている場合、優先サーバーがプライマリサーバーとなり、データベースの所有権を持ちます。優先サーバーとしてマークされているサーバーが使用できなくなると、ミラーサーバーとして稼働していたサーバーがプライマリサーバーとなります。優先サーバーが再開すると、優先サーバーは、まだ所有していないす

べてのトランザクションログエントリを現在のプライマリサーバーから取得します。その後、現在のプライマリサーバーに対して、データベースの所有権の放棄を確認します。優先サーバーと現在のプライマリサーバーはロールを変更し、優先サーバーがプライマリサーバーになり、もう一方のサーバーがミラーサーバーになります。データベースの所有権が変更されると、優先サーバーではないサーバーにあるデータベースへの接続は失われます。

パートナーサーバーを定義する CREATE MIRROR SERVER 文に PREFERRED='YES' を追加することによって、優先サーバーを指定します。次に例を示します。

```
CREATE MIRROR SERVER mirror_server1
AS PARTNER
connection_string='SERVER=mirror_server1;host=localhost:6871'
state_file='c:¥¥server1¥¥server1.state'
preferred='YES';
```

**参照**

- 「CREATE MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』
- 「ALTER MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』
- 「プライマリサーバーのフェールオーバー」1168 ページ
- 「データベースミラーリングモード」1142 ページ

## ミラーサーバーで実行されているデータベースへの読み込み専用アクセスの設定

データベースミラーリングを使用するときは、読み込み専用の接続を使用して、ミラーサーバー上で実行されているデータベースにアクセスできます。この機能は、このデータベースへの読み込み専用アクセスを必要とするレポートなどの操作の負荷を軽減したい場合に役立ちます。

ミラーリングシステムでは、どのデータベースサーバーがプライマリサーバーで、どのデータベースサーバーがミラーサーバーとして動作しているかが不明なこともあります。ミラーサーバー上で実行されているデータベースに接続できるようにするには、タイプ MIRROR を指定してミラーサーバーを定義します。このオプションで、読み込み専用のミラーデータベースへのアクセスに使用するサーバー名を指定すると、接続がミラーサーバーを見つけられるようになります。この方法で指定したサーバー名は、そのデータベースサーバーがデータベースのミラーとして動作している場合のみアクティブになります。たとえば、次の文では、ミラーサーバー上で実行されているデータベースに接続するときに使用する代替サーバー名として mysamplemirror を定義します。

```
CREATE MIRROR SERVER mysamplemirror
AS MIRROR
connection_string='SERVER=mysamplemirror;HOST=winxp-1:6871,winxp-2:6872';
```

データベースに変更を加えようとするとエラーが発生します。これは、-r オプションを使用して読み込み専用でデータベースを起動した場合と同じ動作です。テンポラリテーブルに対して操作を実行することはできますが、ミラーデータベース上ではイベントは起動しません。イベントの起動は、プライマリサーバーからミラーサーバーへのフェールオーバーが発生するまで始まりません。DatabaseStart と MirrorFailover のイベントは (定義されている場合)、その時点で起動します。「システムイベント」1126 ページを参照してください。

フェールオーバーが発生してミラーサーバーがプライマリサーバーになっても、ミラーデータベースへの接続は維持されます。フェールオーバー後は、接続によってデータベースに変更を加えることができます。接続先のデータベースが更新可能かどうかを確認するには、ReadOnly データベースプロパティの値を問い合わせてください。

```
SELECT DB_PROPERTY( 'ReadOnly' );
```

**参照**

- 「CREATE MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』
- 「ALTER MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』
- ReadOnly データベースプロパティ 803 ページ

## ミラーデータベース問い合わせ

ミラーデータベースに対して実行されたクエリは、指定された独立性レベルに応じてロックを設定できます。プライマリサーバーからの操作の適用がロックによって妨害される場合は、ロックを保持している接続のトランザクションがロールバックされ、その接続用の開いたカーソルがある場合は閉じられます。独立性レベル 0 で実行されているアプリケーションは、ローロックを追加しませんが、スキーマロックは取得します。プライマリサーバーからの操作の適用がスキーマロックによって妨害される場合は、ミラーデータベース上のトランザクションがロールバックされます。

データベースの一貫したビューが必要な (そのため、独立性レベル 0 を使用できない) アプリケーションでは、スナップショットアイソレーションの使用を検討する必要があります。そのためには、allow_snapshot_isolation オプションを On に設定します。このオプションはプライマリサーバーとミラーサーバーの両方に対して有効なので、スナップショットアイソレーションに関連するコストを考慮する必要があります。

ミラーデータベースへの接続は、プライマリサーバーに対するトランザクションの影響を受けます。これらの操作は、その後ミラーサーバーによって処理され適用されるからです。プライマリサーバー上の更新がコミットされてから、その更新がミラーサーバー上で使用可能になるまでには、わずかな遅延があります。通常この遅延は短時間ですが、ミラーサーバー上で実行されるデータベースにアクセスするときは、このことを念頭においてください。

**参照**

- 「スナップショットアイソレーション」『SQL Anywhere サーバー SQL の使用法』
- 「allow_snapshot_isolation オプション」590 ページ

# データベースサーバーの強制プライマリサーバー化

プライマリサーバーを強制的にシャットダウンする必要が生じ (サーバーを実行しているコンピューターを置き換える場合など)、ミラーサーバーが他の方法でデータベースの所有権を取得しない場合に、ALTER DATABASE 文を使用してミラーサーバーを強制的にプライマリサーバーにすることができます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

この機能を使用するには、ミラーデータベースサーバー上のユーティリティデータベースに接続する必要があります。ユーティリティデータベースに接続するには、コマンドに -su オプションを指定してミラーサーバーを起動します。次のコマンドにより、データベース *mymirroreddb.db* のミラーサーバーが強制的にプライマリサーバーになります。

```
ALTER DATABASE mymirroreddb FORCE START;
```

FORCE START 句は、現在ミラーサーバーとして動作しているデータベースサーバーに、データベースの所有権の取得を強制します。この文は、ミラーサーバー上のユーティリティデータベースに接続している間に実行する必要があります。「ユーティリティデータベース接続」137 ページを参照してください。

プライマリサーバーからミラーサーバーへのフェールオーバーを強制するには、次の操作を行います。

- プライマリサーバーを停止します。
- プライマリサーバーのデータベースに接続しているときに ALTER DATABASE SET PARTNER FAILOVER を実行します (この文を実行すると、プライマリサーバーでデータベースが再起動され、プライマリサーバーがミラーサーバーになります)。

**参照**

- 「ALTER DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』

## 監視サーバーとしてコピーノードを使用する

新しいデータベースサーバーとして作成して高可用性の読み込み専用スケールアウトシステムの監視サーバーを実行したくない場合は、代わりにシステムの既存のコピーノードを使用することができます。

**♦ 監視サーバーを定義してコピーノードを参照するには、次の手順に従います。**

- 監視サーバーを作成して、その名前が高可用性システムのどのデータベースサーバーの名前とも一致しないようにします。監視サーバーの名前は、ミラーサーバー定義の監視サーバー接続文字列を単独で識別するのに使用されます。

**参照**

- 「CREATE MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』

**例**

たとえば、次のサーバー定義を使用する高可用性および読み込み専用スケールアウトシステムに、TheArbiter という名前のデータベースサーバーはありません。

```
CREATE MIRROR SERVER "scaleout_child"
AS COPY connection_string = 'SERVER=scaleout_child;HOST=winxp-2:6878';
CREATE MIRROR SERVER "TheArbiter"
AS ARBITER connection_string = 'SERVER=scaleout_child;HOST=winxp-2:6878';
```

## プライマリサーバーのフェールオーバー

次の文を実行すると、プライマリサーバーからミラーサーバーへのデータベースミラーリングのフェールオーバーを起動できます。

```
ALTER DATABASE SET PARTNER FAILOVER;
```

この文は、優先サーバーを指定する代わりに使用できます。また、特定のデータベースサーバーに対してデータベースの所有権が指定されているときに制御するロジックで使用することができます。たとえば、パートナーサーバー (PartnerState データベースプロパティの値で決まります) の可用性、またはデータベースに対する接続数 (ConnCount データベースプロパティの値で決まります) に基づいて、フェールオーバーを起動できます。

この文を実行すると、データベースに対する既存の接続は、文を実行した接続も含めて、閉じられます。この文がプロシージャーまたはイベントに含まれている場合、後続する他の文は実行されない可能性があります。この文の実行に必要なパーミッションは、-gk サーバーオプションで制御します。

**参照**

- 「優先データベースサーバー」1164 ページ
- 「ALTER DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「-gk dbeng12/dbsrv12 サーバーオプション」235 ページ
- ConnCount データベースプロパティ 787 ページ
- PartnerState データベースプロパティ 801 ページ

## ミラーリングシステムのデータベースサーバーの停止

高可用性システムのデータベースサーバでメンテナンスを実行する前に、データベースサーバを停止します。

**前提条件**

サーバを停止するにはユーティリティデータベースへの接続を使用する必要があります。このため、データベースサーバの起動時には -su サーバオプションを含めることをおすすめします。

**内容と備考**

次のとおりです。

**♦ データベース停止ユーティリティ (dbstop) を使用して、プライマリサーバ、ミラーサーバ、または監視サーバを停止します。**

- dbstop コマンドを発行して、データベースサーバーを停止します。

  たとえば、次のコマンドを実行するとデータベースサーバー mirror_server1 が停止します。

  ```
  dbstop -c "UID=DBA;PWD=sql;DBN=utility_db;LINKS=tcpip" mirror_server1
  ```

### 結果

データベースサーバが停止します。高可用性システムに 2 台以上のサーバがある場合、ユーザーは、いずれかのデータベースのコピーに接続可能である必要があります。ただし、停止するサーバによって、データベースを更新できたり、できなかったりする場合があります。

### 次の手順

プライマリサーバ、ミラーサーバ、または監視サーバを停止しても、システムは動作し続けますが、高可用性を維持するため、停止したデータベースサーバを再起動するか、置き換える必要があります。

**参照**

- 「ユーティリティデータベース接続」137 ページ

## ミラーサーバーの削除

DROP MIRROR SERVER 文を使用してミラーサーバーを削除できます。データベースミラーリングシステムの一部として実行されているミラーサーバーは削除しないことをおすすめします。

データベースミラーリングシステムを設定するときに、プライマリサーバーになることができるミラーサーバーには、2 つの定義 (CREATE MIRROR SERVER...AS PARTNER と、CREATE MIRROR SERVER...AS PRIMARY または CREATE MIRROR SERVER...AS MIRROR) を指定します。ミラーサーバーの削除理由によっては、ロール定義とパートナーサーバー定義の両方をデータベースから削除する必要がある場合があります。

> **注意**
> 同じミラーサーバー名を保持して設定内容を変更する場合は、CREATE OR REPLACE MIRROR SERVER 文または ALTER MIRROR SERVER 文を使用できます。 「CREATE MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』と「ALTER MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

### ♦ ミラーサーバーの削除 (SQL の場合)

データベースミラーリングシステムの一部として実行されているミラーサーバーは削除しないことをおすすめします。

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. DROP MIRROR SERVER 文を実行してミラーサーバーを削除します。次に例を示します。

```
DROP MIRROR SERVER mirror-server-name;
```

**参照**

- 「DROP MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』

## プライマリサーバー障害

プライマリサーバー障害からのリカバリ手順は、データベースミラーリングシステムで使用している同期実行モードによって異なります。

同期モードで実行している場合は、プライマリサーバーに存在するすべてのトランザクションがミラーサーバーでもコミットされていることが保証されます。ミラーサーバーは、ほとんどすべての場合において、ユーザーによる介入なしに新しいプライマリサーバーになることができます。

**注意**
まれに、チェックポイントの実行中にプライマリサーバーがミラーサーバーと監視サーバーの両方との接続を失った場合は、次にこれら 2 つのサーバーが接続したときに同期に失敗することがあります。ミラーサーバーは、データベースにプライマリサーバーとの互換性がないためシャットダウンされることを示すエラーを報告します。この場合は、ミラーサーバーが正常に起動して同期するように、現在のプライマリサーバーのデータベースをミラーサーバーに手動でコピーまたはバックアップする必要があります。

非同期モードおよび非同期フルページモードでは、プライマリサーバーに適用されたコミット済みトランザクションがミラーサーバーにすべて適用されているとは限らないので、プライマリサーバーからミラーサーバーへのフェールオーバーは自動ではありません。このため、自動フェールオーバーの実行が指定されていないかぎり、非同期のどちらかのモードを使用する場合は、ミラーサーバーはデフォルトで、プライマリに障害が発生してもデータベースの所有権を取得できません。障害が発生したサーバーは、再起動後にトランザクションが失われたかどうかを確認します。トランザクションが失われていた場合、データベースサーバーメッセージログにメッセージを書き込んで、データベースをシャットダウンします。

元のミラーサーバーを新しいプライマリサーバーとして起動する場合、両方のサーバー上のデータベースファイルを同じ状態にする方法が 2 つあります。

- データベースとトランザクションログファイルを元のプライマリサーバーからミラーサーバーにコピーし、ミラーサーバーを新しいプライマリサーバーとして起動します。ALTER DATABASE 文を使用すると、サーバーを強制的にプライマリサーバーにできます。

- 元のミラーサーバーで、dbbackup を使用してバックアップを実行します。元のプライマリサーバーにファイルをコピーし、データベースサーバーを起動します。「バックアップユーティリティ (dbbackup)」913 ページを参照してください。

**参照**

- 「ALTER DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』

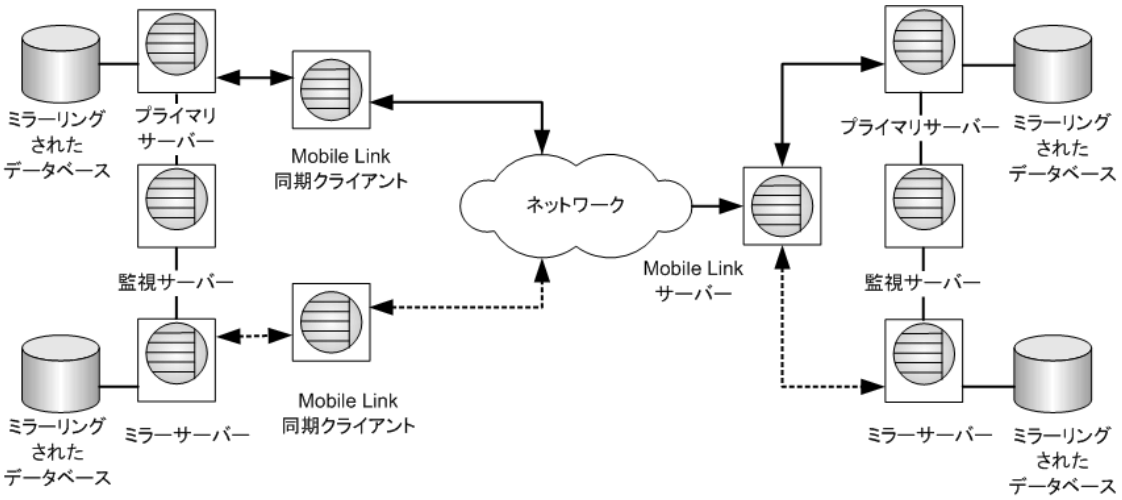
## データベースミラーリングと Mobile Link

この項では、SQL Anywhere 統合データベースを複数の SQL Anywhere リモートデータベースとともに使用して、高可用性環境内で Mobile Link 同期を実行するための設定方法について説明します。Mobile Link 同期システムは、高可用性環境内で統合データベースを実行するように設定

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

することも、リモートデータベースを高可用性環境内で実行するように設定することも、またはその両方に設定することもできます。

次の図は、統合データベースとリモートデータベースの両方がミラーリングされている環境で実行される Mobile Link 同期を示しています。




## SQL Anywhere 統合データベースの高可用性

「プライマリサーバー」は、データベースファイルとトランザクションログのアクティブなコピーを管理します。このサーバーは、アクティブな接続を受け入れて、データベーストランザクションを処理します。すべてのトランザクションはプライマリサーバーによってミラーサーバーに送信されます。

「ミラーサーバー」は、プライマリサーバーに適用されたすべてのトランザクションを、プライマリサーバーから直接受信します。フェールオーバーが発生すると、ミラーサーバーがプライマリサーバーになります。

「Mobile Link サーバー」は、プライマリサーバーとの ODBC 接続を確立し、Mobile Link クライアントからのアップロードストリームを処理します。また、クライアントに送信するダウンロードストリームを生成します。Mobile Link サーバーは、プライマリとして動作するサーバーに常に接続し、ミラーサーバーには接続しません。

## SQL Anywhere リモートデータベースの高可用性

データの同期は、「Mobile Link クライアント」がリモートデータベースと Mobile Link サーバーとの接続を確立したときに開始されます。Mobile Link クライアントは、アクティブなトランザクションログとオフライントランザクションログをスキャンする必要があります。ただし、高可用性環境では、どのデータベースが現在プライマリサーバーとして動作しているかや、アクティブとオフラインのトランザクションログがどこにあるかが常にわかっているとはかぎりません。オフライントランザクションログがデータベースに関連付けられている場合は、Mobile Link クライアントを起動するときに **-e dir=***path-name* を指定する必要があります。「OfflineDirectory (dir) 拡張オプション」『Mobile Link クライアント管理』を参照してください。

Mobile Link クライアントは、リモートデータベースのトランザクションログに基づいて、Mobile Link サーバーに送信するためのアップロードストリームを生成します。その後クライアントはダウンロードストリームを処理して適用します。

### プライマリサーバー障害からのリカバリ

高可用性データ同期システムに単一障害点ができないようにするために、複数の Mobile Link サーバーと負荷分散装置を使用することをおすすめします。これにより、データ同期環境の各コンポーネントが冗長化され、環境内でハードウェアやソフトウェアの障害が発生した場合でもシステムが機能します。 複数の Mobile Link サーバーの使用に関する詳細については、「Mobile Link サーバーのサーバーファームでの実行」『Mobile Link サーバー管理』と「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

同様に、リモートデータベースでの単一障害点を防止するには、xp_cmdshell システムプロシージャーのような外部プロシージャーを Mobile Link クライアントの呼び出しに使用する実装を、プライマリデータベースとミラーデータベースの両方に格納されているスケジュールされたイベントによってトリガーできます。

### 参照

- 「xp_cmdshell システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「SQL Anywhere の高可用性」1136 ページ
- 「Mobile Link サーバーテクノロジ」『Mobile Link サーバー管理』

## データベースミラーリングとトランザクションログファイル

パートナーサーバーは、起動されると、現在のトランザクションログファイルが存在するディレクトリ内のすべてのトランザクションログファイルを調べて、適用する必要があるファイルを特定します。データベースサーバーは、トランザクションログに記録された操作をデータベースに適用してから、プライマリサーバーとミラーサーバーのどちらとして動作するかを判断します。トランザクションログファイルとデータベースファイルは同一のディレクトリに格納できます。ただし、このディレクトリは、データベースの起動を遅延させる可能性があるので、他のファイルは入れるべきではありません。

プライマリサーバーロールを引き受けたサーバーは、ミラーサーバー上の現在のトランザクションログと同一の開始オフセットのトランザクションログを持ち、同様に、その後のトランザクションログファイルがあればプライマリサーバーの現在のトランザクションログファイルまでのものを持つ必要があります。

ミラーサーバーロールを引き受けたサーバーは、プライマリサーバーからトランザクションログページの受信を開始します。プライマリでトランザクションログの名前が変更された場合は、ミラーでも名前が変更されます。ミラーは、新しいトランザクションログページを、トランザクションログ用に指定された名前の新しいファイルに書き出します。

プライマリでは、トランザクションログファイルを定期的に削除できます。トランザクションログファイルの名前が変更されるたび、ミラーには、プライマリで削除されずに残っている最も古いトランザクションログファイルがどれであるかが通知されます。ミラーでも、通知されたものより古いトランザクションログファイルはすべて削除されます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

トランザクションログのトランケートを要求するプライマリサーバーに対してバックアップを実行している間は、ミラーサーバーが使用できなくなるので、プライマリでのトランザクションログの削除は、トランザクションログのトランケートとは異なる方法で実行する必要があります (たとえば、xp_cmdshell を使用して作成後 1 週間が過ぎたファイルを削除するイベントをスケジュールすることで実行します)。

## データベースミラーリングにおけるシステムイベント

データベースミラーリングでサポートされているシステムイベントは次のとおりです。

- **MirrorFailover**　このイベントは、データベースサーバーがミラーリングされたデータベースの所有権を取得するたびに起動します。たとえば、初めてサーバーが起動され、データベースを所有する必要があると判断した場合に起動します。また、前回ミラーとして動作していたサーバーが、プライマリサーバーがダウンしており、監視サーバーに確認した結果、所有権を取得する必要があると判断した場合にも起動します。

- **MirrorServerDisconnect**　プライマリサーバーとミラーサーバーまたは監視サーバーとの接続が失われると、MirrorServerDisconnect イベントが起動します。このイベントのハンドラー内部で、EVENT_PARAMETER ('MirrorServerName') の値は接続が失われたサーバーの名前です。

イベントは、現在ミラーサーバーとして動作しているサーバーでは起動しません。また、ミラーリングイベントを監視サーバー上で実行するよう定義することはできません。なぜなら、イベントは定義されているデータベースのコンテキストでのみ実行されますが、監視サーバーはミラーリングされているデータベースのコピーを使用しないからです。

上記のイベントは、ミラーデータベースに対するアクションの必要性を電子メールで通知するためのメカニズムとして使用できます。ただし、プライマリサーバー上で実行されているデータベースが使用できなくなるあらゆる状況でこれらのイベントが起動するわけではありません。たとえば、プライマリサーバーとミラーサーバーの両方に影響する停電があった場合、どちらのイベントも起動しません。このような監視が必要な場合は、別のコンピューターでスクリプト言語を使用することで実装できます。たとえば、dbping を定期的に呼び出してミラーデータベースに接続します。 「Ping ユーティリティ (dbping)」994 ページを参照してください。

次の例は、フェールオーバーの発生を管理者に通知するイベントを作成しています。

```
CREATE EVENT mirror_server_unavailable
TYPE  MirrorServerDisconnect
HANDLER
BEGIN
 CALL xp_startmail ( mail_user ='George Smith',
                     mail_password ='mypwd' );
 CALL xp_sendmail( recipient='DBAdmin',
    subject='Mirror server disconnect occurred',
    "message"='The following server is unavailable in the mirroring system: '
    || event_parameter( 'MirrorServerName' ) );
 CALL xp_stopmail ( );
END;
```

**参照**

- 「システムイベント」1126 ページ

## データベースミラーリングとパフォーマンス

プライマリサーバーとミラーサーバーを実行する各コンピューターは、ハードウェア構成 (プロセッサー、ディスク、メモリなど) を同じにしておくのが理想的です。どちらかのコンピューターで実行されているデータベースサーバーは、常にミラーリング対象のデータベースのプライマリサーバーとして動作する可能性があります。プライマリでの更新アクティビティによっては、ミラーサーバーの使用率か低くなるのが一般的です。

データベースを更新するトランザクションのパフォーマンスは、トランザクションのサイズとコミットの頻度の影響を受けます。トランザクションのサイズまたはコミットの頻度のいずれかを増やすと、プライマリサーバーがミラーサーバーとやり取りをする回数が多くなり、それによってプライマリサーバーが遅延し、ユーザーにそれらの遅延を通知する可能性が高くなります。

プライマリサーバーに対するクエリのパフォーマンスは、通常はミラーリングの影響を受けません。アプリケーションが更新よりも多くの読み込みを行い、かつ監査が有効になっていない場合は、プライマリサーバーのパフォーマンスでの変化を経験することはほとんどありません。しかし、データベース監査が有効になっている場合、監査がトランザクションログに追加の情報を書き込むため、パフォーマンスの変化を経験する可能性があります。

非同期モードまたは非同期フルページモードのいずれかで実行しているミラーリングシステムは、プライマリサーバーに対して同期モードで動作しているときよりも待機する機会をより少なくするように要求します。しかし、非同期モードまたは非同期フルページモードであっても、プライマリサーバーからミラーサーバーに送信されるメッセージのうちのいくつかは、ミラーからの受信確認を待機するので、これが遅延を引き起こすこともあります。さらに、パケットをミラーに送信するときには少量のオーバーヘッドも存在します。

クライアントアプリケーションのパフォーマンスは、ミラーリングシステムのサーバー間でのネットワーク接続によって、特に、サーバーが地理的に異なる場所にあるときに、影響を受ける可能性があります。サーバー間の低速のネットワーク接続は、クライアントアプリケーションのパフォーマンスを低下させる可能性があります。

**参照**

- 「データベースミラーリングモード」1142 ページ

## データベースミラーリングとバックアップ

データベースミラーリングはデータ消失のリスクを最低限に抑えるのに役立ちますが、データベースミラーリングシステムに参加しているデータベースのバックアップと検証も実施することをおすすめします。

BACKUP DATABASE 文を使用すると、データベースサーバーを基準にしてバックアップを実行できます。BACKUP DATABASE 文はプライマリデータベースサーバー上で実行されるので、指定するファイル名には、プライマリおよびミラーのデータベースサーバー両方で一貫性のあるネットワークドライブまたは UNC 名を指定する必要があります。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

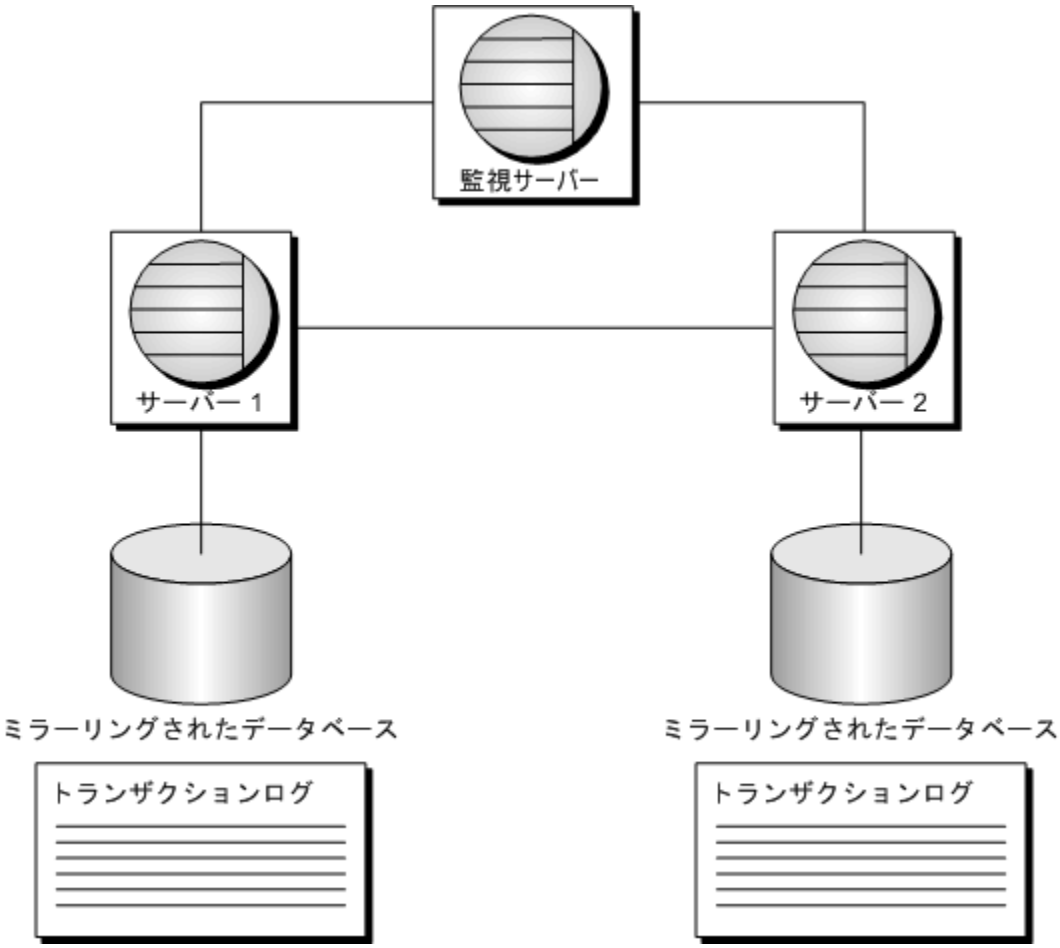
別の方法として、dbbackup ユーティリティを使用してクライアント側のバックアップを実行することもできます。

**参照**

- 「BACKUP 文」『SQL Anywhere サーバー SQL リファレンス』
- 「バックアップユーティリティ (dbbackup)」913 ページ
- 「バックアップとデータリカバリ」1071 ページ
- 「データベースの検証」1117 ページ

## データベースミラーリングのシナリオ

以下のシナリオは、ミラーリングシステムでサーバーが使用できなくなった場合に何が起こるかを理解するのに役立ちます。ここで使用するデータベースミラーリングは、同期モードで実行されるサーバー 1、サーバー 2、監視サーバーで構成されています。



ミラーリングシステム内のデータベースサーバーのステータスは、MirrorState、PartnerState、ArbiterState の各データベースプロパティを使用していつでも確認できます。MirrorState データベースプロパティ 799 ページ、PartnerState データベースプロパティ 801 ページ、ArbiterState データベースプロパティ 782 ページを参照してください。

プライマリ、ミラー、監視サーバー間の通信は、ミラーリングシステム内で定期的にチェックされます。これら 3 つのサーバー間の通信の中断は、1 つ以上のサーバーが利用できなくなることや、サーバー間のネットワークの停止によって引き起こされることがあります。

### シナリオ 1：プライマリサーバーが使用できなくなった場合

1. プライマリサーバー (サーバー 1) が使用できなくなります。クライアントはすべて切断されます。

2. 監視サーバーとサーバー 2 はサーバー 1 が使用できないことを検知します。

3. 監視サーバーとサーバー 2 はクォーラムを形成し、サーバー 2 がプライマリサーバーになります。

4. サーバー 2 は、クライアント接続の受け付けを開始します。

このシナリオにおいて、非同期モードまたは非同期フルページモードを使用し、自動フェールオーバーの実行を指定していない場合、クライアントが再び接続できるようにするには、データベースのコピーを作成し、稼働しているサーバーを再起動する必要があります。

プライマリサーバーが使用できなくなったときのリカバリの詳細については、「プライマリサーバー障害」1170 ページを参照してください。

### シナリオ 2：プライマリサーバーが使用できなくなり、再起動された場合

1. プライマリサーバー (サーバー 1) が使用できなくなります。クライアントはすべて切断されます。

2. 監視サーバーとミラーサーバー (サーバー 2) が、プライマリサーバー (サーバー 1) が使用できなくなったことを検知します。

3. 監視サーバーとサーバー 2 はクォーラムを形成し、サーバー 2 がプライマリサーバーになります。

4. サーバー 2 は、クライアント接続の受け付けを開始します。

5. サーバー 1 が再びオンラインになり、サーバー 2 と監視サーバーに再接続します。

6. サーバー 1 がクォーラムを要求しますが、サーバー 2 がすでにプライマリサーバーになっています。

7. サーバー 1 はミラーサーバーになって、サーバー 2 からの変更を待機します。

8. サーバー 2 は変更をサーバー 1 に送信します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

サーバー 1 がサーバー 2 からすべてのトランザクションを受信する前にサーバー 2 が使用できなくなった場合、サーバー 1 は同期された状態にはなれません。サーバー 2 が再び使用可能になるのを待って、まだ取得していないトランザクションを取得して適用する必要があります。

プライマリサーバーが使用できなくなったときのリカバリの詳細については、「プライマリサーバー障害」1170 ページを参照してください。

**シナリオ 3：ミラーサーバーが使用できなくなった場合**

1. ミラーサーバー (サーバー 2) が使用できなくなります。
2. 監視サーバーとサーバー 1 はミラーサーバー (サーバー 2) が使用できないことを検知します。

   クライアント接続は影響を受けません。引き続きプライマリサーバーに接続できます。ただし、サーバー 1 または監視サーバーが使用できなくなった場合、クライアントは接続できなくなります。

**シナリオ 4：ミラーサーバーが使用できなくなり、再起動された場合**

1. ミラーサーバー (サーバー 2) が使用できなくなります。
2. 可用性は変わらないため、クライアント接続は影響を受けません。引き続きプライマリサーバーに接続できます。ただし、サーバー 1 または監視サーバーが使用できなくなった場合、クライアントは接続できなくなります。
3. サーバー 2 が再びオンラインになり、サーバー 1 と監視サーバーに再接続します。
4. サーバー 2 がクォーラムを要求しますが、サーバー 1 がすでにプライマリサーバーになっています。
5. サーバー 2 はミラーサーバーになって、サーバー 1 からの変更を待機します。
6. サーバー 1 は変更をサーバー 2 に送信します。

   可用性は変わらないため、クライアント接続は影響を受けません。引き続きサーバー 1 に接続します。

**シナリオ 5：監視サーバーが使用できなくなった場合**

1. サーバー 1 (プライマリサーバー) とサーバー 2 (ミラーサーバー) が、監視サーバーがダウンしたことを検知します。
2. どちらのサーバーも引き続き使用できます。クライアント接続は切断されません。

   監視サーバーが再びオンラインになると、サーバー 1 とサーバー 2 はそれを検知して、監視サーバーとの通信を開始します。クライアントにとっては、データベースの可用性に変化はありません。

   監視サーバーがない状態でサーバー 1 またはサーバー 2 が使用できなくなった場合、もう一方のサーバーは単独ではクォーラムを満たすことができないので、データベースは使用できなくなります。

### シナリオ 6 : 監視サーバーが再起動された場合

1. 監視サーバーが再びオンラインになり、サーバー 1 とサーバー 2 に再接続します。

   可用性は変わらないため、クライアント接続は影響を受けません。

## データベースミラーリングの切断された接続のシナリオ

データベースミラーリングシステムでは、データベースサーバー間の接続が切断されると、次の注意事項が当てはまります。

- 切断されたネットワーク接続を通知するためにサーバーに必要な時間は、次の状況によって異なります。

  1. 接続がどのように切断されるのか。ネットワークケーブルがコンピューターに差し込まれていない場合、オペレーティングシステムの中にはこの問題を即座に検出して、接続を切断させるものもあります。ネットワークスイッチの障害によって問題が引き起こされた場合は、問題がサーバーによって即座に検出される可能性はそれよりも少なくなります。

  2. ミラー接続文字列で指定された活性タイムアウト設定。LivenessTimeout 接続プロパティ 728 ページを参照してください。

- プライマリサーバーがミラーリングされたデータベースの所有権を維持するには、少なくとも他のサーバー (ミラーまたは監視サーバー) の 1 つに接続される必要があります。他のサーバーのどちらに対する接続も失われた場合、データベースは再起動され、プライマリサーバーは、それらの接続の 1 つに少なくとも再確立できて、そのステータスを検証できるまで待ちます。

- 接続の切断を引き起こした問題が修正されたときに、サーバー間の接続は自動的に再確立されます。

- サーバーは、切断した接続を検出するまで、これまで割り当てられていたロールを維持します。この動作は、結果として、これまでミラーサーバーであったサーバーがプライマリサーバーへの切断された接続を検出した後で、プライマリサーバーがミラーサーバーへの接続が切断されたことを検出した場合、短い期間だけ 2 つのサーバーがプライマリサーバーとして動作しようとすることになります。元のプライマリサーバーで行われた最初の更新は、サーバーが他のサーバーにその変更を送信しようとした時点でブロックされます。これによって、この状況が発生したためにトランザクションが失われる可能性が限定されます。

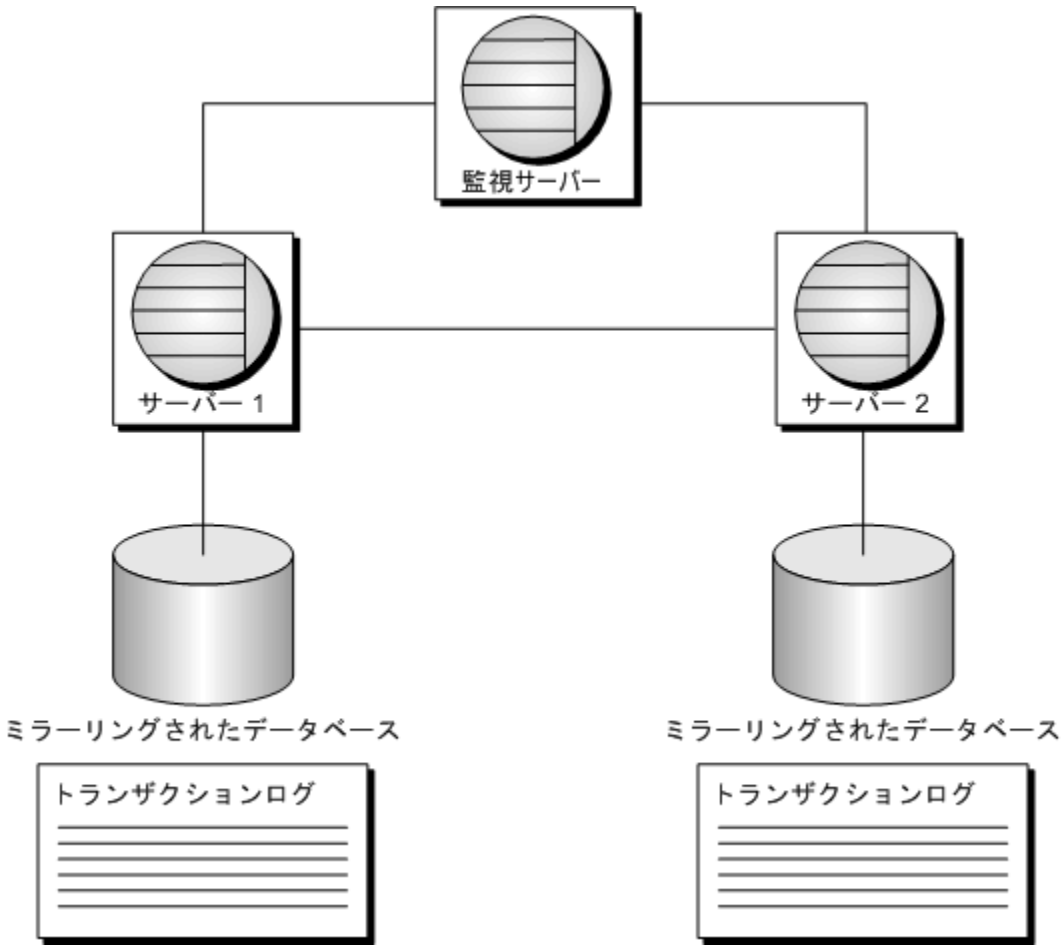
  元のプライマリは、元のミラーサーバーからの切断された接続を検出し、監視サーバーのステータスをチェックしたときに、フェールオーバーが発生していることを確認し、データベースを再起動します。

- ミラーとして動作しているサーバーがデータベースを再起動する必要がある場合、そのサーバー上のデータベースへのアクセスは、プライマリサーバーへの接続が再確立されるまで許可されません。

- ネットワークの障害は、ネットワークのトポロジによってはクライアント接続とミラーサーバー接続の両方に影響する可能性があります。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

以下のシナリオは、ミラーリングシステムで接続が切断された場合に何が起こるかを理解するのに役立ちます。ここで使用するデータベースミラーリングは、同期モードで実行されるサーバー 1、サーバー 2、監視サーバーで構成されています。




ミラーリングシステム内のデータベースサーバーのステータスは、MirrorState、PartnerState、ArbiterState の各データベースプロパティを使用していつでも確認できます。MirrorState データベースプロパティ 799 ページ、PartnerState データベースプロパティ 801 ページ、ArbiterState データベースプロパティ 782 ページを参照してください。

### シナリオ 7：プライマリサーバーと監視サーバーの接続が切断された場合

1. プライマリサーバー (サーバー 1) と監視サーバーとの接続が切断されます。
2. サーバー 1 と監視サーバーは、互いの接続が失われたことを検出します。
3. プライマリサーバーとミラーサーバーの両方は利用可能なままです。クライアント接続は切断されません。サーバー 1 は引き続き変更をサーバー 2 に送信します。

- サーバー 1 が利用不可能になったか、ミラーサーバー (サーバー 2) への接続を失っている場合、サーバー 2 は自身が監視サーバーにまだ接続されていることを前提としてプライマリサーバーになります。

- サーバー 1 がサーバー 2 および監視サーバーへの接続を失っている場合、サーバー 1 上のデータベースは使用できなくなります。サーバー 1 は、データベースを再起動し、監視サーバーとサーバー 2 への再接続が確立するのを待ちます。

- サーバー 1 とサーバー 2 が、どちらも監視サーバーへの接続を失い、その一方で互いへの接続は続いている場合、その動作は「シナリオ 5：監視サーバーが使用できなくなった場合」で説明したものと同じになります。

**シナリオ 8：ミラーサーバーと監視サーバーの接続が切断された場合**

1. ミラーサーバー (サーバー 2) と監視サーバーとの接続が切断されます。

2. サーバー 2 と監視サーバーは、互いの接続が失われたことを検出します。

3. プライマリサーバーとミラーサーバーの両方は利用可能なままです。クライアント接続は切断されません。

4. サーバー 1 は引き続き変更をサーバー 2 に送信します。

5. サーバー 1 とサーバー 2 の両方が監視サーバーの再接続を待機します。

   - サーバー 2 が利用不可能になるか、サーバー 1 への接続を失う場合、サーバー 1 へのクライアント接続は、可用性の変更がないため、影響を受けません。サーバー 2 へのクライアント接続は失われます。

   - サーバー 1 とサーバー 2 が、どちらも監視サーバーへの接続を失い、その一方で互いへの接続は続いている場合、その動作は「シナリオ 5：監視サーバーが使用できなくなった場合」で説明したものと同じになります。

6. 監視サーバーが再びオンラインになり、サーバー 1 とサーバー 2 は監視サーバーに接続します。

**シナリオ 9：プライマリサーバーとミラーサーバーの接続が切断された場合**

1. プライマリサーバー (サーバー 1) とミラーサーバー (サーバー 2) の接続が切断されます。

2. サーバー 1 とサーバー 2 は、接続が失われたことを検出しますが、異なるタイミングで検出する可能性があります。どちらのサーバーも監視サーバーのステータスを確認してから、監視サーバーからの応答に基づいて自身のアクションを実行します。

   - サーバー 1 が、接続の切断を先に検出した場合、監視サーバーのステータスを更新して、自身が引き続きプライマリサーバーであることと、サーバー 2 のデータベースが最新ではない可能性があることを示します。サーバー 2 は、監視サーバーを確認したときに、サーバー 1 が引き続きプライマリであることを理解します。サーバー 2 は自身のデータベースを再起動します。サーバー 2 はサーバー 1 への接続が再確立するのを待ってから、データベースへのアクセスを許可します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

- サーバー 2 が、接続の切断を先に検出した場合、監視サーバーのステータスを更新して、フェールオーバーが発生したことと、サーバー 2 がプライマリサーバーになったことを示します。サーバー 1 は、監視サーバーのステータスを確認して、フェールオーバーが発生したことと、サーバー 2 がプライマリサーバーになったことを理解します。サーバー 1 はデータベースを再起動し、サーバー 2 への接続が再確立するのを待ってから、データベースへのアクセスを許可します。

**シナリオ 10 : プライマリサーバーから、ミラーサーバーおよび監視サーバーへの接続が切断された場合**

1. プライマリサーバー (サーバー 1) とミラーサーバー (サーバー 2) の間の接続が切断し、プライマリサーバー (サーバー 1) と監視サーバーとの接続が切断します。

2. サーバー 1 は監視サーバーとサーバー 2 が使用できなくなったことを検出します。サーバー 1 はデータベースを再起動して、他のサーバーへの再接続が確立するのを待ちます。クライアントはすべて切断されます。

3. 監視サーバーとサーバー 2 はサーバー 1 が使用できないことを検出します。

4. 監視サーバーとサーバー 2 はクォーラムを形成し、サーバー 2 がプライマリサーバーになります。

   監視サーバとサーバ 2 の動作は、「シナリオ 1 : プライマリサーバーが使用できなくなった場合」で説明したものと同じです。

5. サーバー 2 は、クライアント接続の受け付けを開始します。

6. サーバー 1 が再びオンラインになり、サーバー 2 と監視サーバーに再接続します。

7. サーバー 1 はミラーサーバーになって、サーバー 2 からの変更を待機します。

8. サーバー 2 は変更をサーバー 1 に送信します。

**シナリオ 11 : ミラーサーバーから、プライマリサーバーおよび監視サーバーへの接続が切断された場合**

1. ミラーサーバー (サーバー 2) およびプライマリサーバー (サーバー 1) の間の接続と、ミラーサーバー (サーバー 2) と監視サーバーの接続が切断します。

2. サーバー 2 は監視サーバーとサーバー 1 が使用できなくなったことを検出します。サーバー 2 は、データベースを再起動し、プライマリサーバーと監視サーバーへの再接続が確立するのを待ちます。サーバー 2 へのクライアント接続はすべて切断されます。

   プライマリサーバーへのクライアント接続は影響を受けません。クライアントは引き続きプライマリサーバーに接続できます。

3. 監視サーバーとサーバー 1 はサーバー 2 が使用できないことを検出します。

   監視サーバとサーバ 1 の動作は、「シナリオ 3 : ミラーサーバーが使用できなくなった場合」で説明したものと同じです。

4. サーバー 1 は監視サーバーのステータスを更新して、サーバー 2 が最新ではなくなったことを示します。
5. サーバー 2 が再びオンラインになり、サーバー 1 と監視サーバーに再接続します。
6. サーバー 2 はミラーサーバーのロールを再開し、サーバー 1 からの変更を待機します。
7. サーバー 1 は変更をサーバー 2 に送信します。

**シナリオ 12 : 監視サーバーから、プライマリサーバーおよびミラーサーバーへの接続が切断された場合**

1. 監視サーバーとプライマリサーバーとの接続、および監視サーバーとミラーサーバー (サーバー 2) との接続が切断します。
2. 監視サーバーへの接続が失われた場合、動作は、「シナリオ 5 : 監視サーバーが使用できなくなった場合」で説明したものと同じです。
3. 接続が復元された場合、動作は、「シナリオ 6 : 監視サーバーが再起動された場合」で説明したものと同じです。

**シナリオ 13 : すべてのサーバー間の接続が切断された場合**

1. すべてのサーバー間の接続が切断します。
2. プライマリサーバー (サーバー 1)、ミラーサーバー (サーバー 2)、監視サーバーは、互いのサーバーへの接続が失われたことを検出します。
3. サーバー 1 は、すべてのクライアントへの接続を切断してデータベースを再起動します。
4. サーバー 2 は、すべてのクライアントへの接続を切断してデータベースを再起動します。
5. サーバー 1 とサーバー 2 は、互いへの再接続を待機して、そのロールを決定します。
6. 接続が再確立されると、サーバー 1 とサーバー 2 は自身のステータスと相手のステータスに基づいて自らのロールを決定します。

# SQL Anywhere Veritas Cluster Server エージェント

**注意**
SQL Anywhere Veritas Cluster Server エージェントには、別途ライセンスが必要です。「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

「クラスタ」とは、一連のアプリケーションを実行するために連携して動くコンピューター (「ノード」と呼ばれる) によるグループです。クラスタ上で実行されているアプリケーションに接続するクライアントは、クラスタを単一のシステムとして扱います。あるノードに障害が発生すると、クラスタ内の別のノードが、障害のあったノードが提供するサービスを自動的に引き継ぎます。クライアント側では、可用性がわずかに悪化したように見えることがありますが (残りのノードでサービスを再開するまでの時間)、ノードに障害が発生したことまではわかりません。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

SQL Anywhere でクラスタリングを使用する場合、データベースまたはデータベースサーバーがクラスタ内の別のノードにフェールオーバーされると、コミットされていないトランザクションはすべて失われます。また、フェールオーバーが発生した場合は、クライアントはデータベースに再接続する必要があります。

SQL Anywhere ではさまざまなクラスタ環境をサポートしています。そのような環境では、クラスタソフトウェアが任意のアプリケーションを自動フェールオーバーの対象となる汎用リソースにすることで、高可用性が実現されます。ただし、フェールオーバーできるのはデータベースサーバープロセスのみで、監視プロセスや制御プロセスについては制限があります。

詳細については、http://www.sybase.com/detail?id=1034743 を参照してください。

ほとんどのクラスタソフトウェアには、特定のアプリケーション用のカスタムリソースを作成するための API が用意されています。SQL Anywhere には、Veritas Cluster Server 用のカスタムフェールオーバーリソースとして、SAServer と SADatabase の 2 つが用意されています。SAServer エージェントはデータベースサーバーのフェールオーバーを担当し、SADatabase エージェントは個々のデータベースファイルのフェールオーバーを担当します。エージェントは、アプリケーションに応じて、いずれか一方または両方とも使用できます。

SQL Anywhere Veritas Cluster Server エージェントを使用するには、次のようにシステムを設定する必要があります。

- Veritas Cluster Server 4.1 以降を使用する
- クラスタ内の各システムに SQL Anywhere を同じ構成でインストールする
- データベースファイルが、クラスタ内のすべてのシステムからアクセスできる共有記憶装置に格納されている
- ユーティリティデータベースのパスワードが、クラスタ内のすべてのシステムで同じである

  SADatabase エージェントはユーティリティデータベースを使用して、特定のデータベースファイルを起動および停止します。クラスタに属しているすべてのシステムには、同じユーティリティデータベースパスワードを使用する必要があります。ユーティリティデータベースのパスワードを設定するには、データベースサーバーの起動時に -su サーバーオプションを指定します。「-su dbeng12/dbsrv12 サーバーオプション」276 ページを参照してください。

  UNIX の場合、VCS エージェントは *$SQLANY12/vcsagent/saserver* にインストールされます。

新しいエージェントを設定して Veritas Cluster Server に追加するには、次の 3 つの方法があります。

1. Cluster Manager を使用する。
2. コマンドラインユーティリティを使用する。
3. テキストエディターを使用して、*main.cf* 設定ファイルを編集する。

ここでは、Cluster Manager を使用する手順について説明します。

使用可能なユーティリティの詳細については、『**Veritas Cluster Server Administration Guide**』を参照してください。

テキストエディターを使用して *main.cf* を手動で設定する場合は、*main.cf* ファイルを編集する前にすべての Veritas Cluster Server サービスを停止する必要があります。そうしないと、変更が反映されません。

## SAServer エージェント設定

SAServer エージェントは、SQL Anywhere データベースサーバーによるクラスタ内の別のノードへのフェールオーバーを制御します。

**♦ SAServer エージェントの設定**

1. クラスタの各ノードで実行されている SQL Anywhere データベースサーバーをすべてシャットダウンします。
2. クラスタからノードを選択して、*SAServer* というディレクトリをそのノードの *%VCS_HOME%¥bin* ディレクトリの下に作成します。他の Veritas Cluster Server エージェントがこのフォルダーに作成されます (NIC や IP など)。
3. 次のファイルを *%SQLANY12%¥VCSAgent¥SAServer* ディレクトリから手順 2 で作成した *SAServer* ディレクトリにコピーします。
   - *Online.pl*
   - *Offline.pl*
   - *Monitor.pl*
   - *Clean.pl*
   - *SAServer.xml*
4. ファイル *%VCS_HOME%¥bin¥VCSdefault.dll* を *%VCS_HOME%¥bin¥SAServer* ディレクトリにコピーし、名前を *SAServer.dll* に変更します。
5. ファイル *%SQLANY12%¥VCSAgent¥SAServer¥SAServerTypes.cf* を *%VCS_HOME%¥conf¥config* ディレクトリにコピーします。
6. クラスタ内のその他すべてのノードについて、手順 1 ～ 5 を繰り返します。
7. Veritas Cluster Server Manager を起動し、ユーザー名とパスワードを入力してクラスタに接続します。
8. 次の手順で SAServer エージェントを追加します。
   a. **[File]** » **[Import Types]** をクリックします。
   b. *%VCS_HOME%¥conf¥config¥SAServerTypes.cf* に移動して、**[Import]** をクリックします。

**♦ SAServer エージェントを使用してフェールオーバーするデータベースサーバーの設定**

1. Veritas Cluster Server Manager を起動し、ユーザー名とパスワードを入力して接続します。
2. 次の手順で、SAServer をリソースとしてサービスグループに追加します。
   a. **[Edit]** » **[Add]** » **[Resource]** をクリックします。

b. **[Resource Type]** リストから、**[SAServer]** をクリックします。

Windows では、**[Resource Type]** リストで Windows の下に SAServer が表示されない場合、*SAServer.xml* ファイルを *%VCS_ROOT%¥cluster manager¥attrpool¥Win2K¥400* に追加して、クラスタサービスを再起動する必要があります。

c. **[Resource Name]** フィールドに名前を入力します。

d. 以下の属性に次のように属性値を追加します。

- **cmdStart** dbsrv12 -x tcpip *database-file-on-shared-disk* -n *server-name*
- **cmdMonitor** dbping -c "Server=*server-name*"
- **cmdStop** dbstop -c *user-id*,*password* -y

e. **[Enabled]** をクリックします。

これにより、リソースの使用準備が整ったことを示します。

f. **[OK]** をクリックします。

3. リソースの依存性が適切に設定されていることを確認します。共有ディスクリソースや IP アドレスリソースなど、SAServer を起動するために起動してグループ化しておく必要があるリソースが他にも存在します。

4. サービスグループを右クリックして、**[Online]** » [*node-name*] をクリックします。*node-name* は、リソースの実行に使用する、クラスター内のコンピューターの名前です。

これで、サービスグループがオンラインになります。

## SAServer エージェントのテスト

### ♦ SAServer エージェントのフェールオーバーのテスト

1. Interactive SQL からデータベースに接続します。次に例を示します。

```
dbisql -c "UID=DBA;PWD=sql;Server=VCS;LINKS=tcpip"
```

2. 次のクエリを実行します。

```
SELECT * FROM Departments;
```

このクエリはエラーなく実行されます。

3. データベースサーバーを実行しているシステムをシャットダウンします。

これにより、フェールオーバーが発生し、すべてのリソースが代替サーバーで起動されます。

4. 同じ接続文字列を使用して Interactive SQL から再接続し、クエリを再度実行します。接続もクエリの実行も正常に行われます。

# SADatabase エージェントの設定

SADatabase エージェントは、SQL Anywhere データベースによるクラスタ内の別のノードへのフェールオーバーを制御します。

### ♦ SADatabase エージェントの設定

1. クラスタの各ノードで実行されている SQL Anywhere データベースサーバーをすべてシャットダウンします。

2. クラスタ内のいずれか 1 つのノードに *%VCS_HOME%¥bin¥SADatabase* というディレクトリを作成します。

3. 次のファイルを *%SQLANY12%¥VCSAgent¥SADatabase* ディレクトリから、手順 2 で作成した *%VCS_HOME%¥bin¥SADatabase* ディレクトリにコピーします。
   - *Online.pl*
   - *Offline.pl*
   - *Monitor.pl*
   - *Clean.pl*
   - *SADatabase.xml*

4. ファイル *%VCS_HOME%¥bin¥VCSdefault.dll* を *%VCS_HOME%¥bin¥SADatabase* ディレクトリにコピーし、名前を *SADatabase.dll* に変更します。

5. ファイル *%SQLANY12%¥VCSAgent¥SADatabase¥SADatabaseTypes.cf* を *%VCS_HOME%¥conf¥config* ディレクトリにコピーします。

6. クラスタに属するその他すべてのシステムについて、手順 1 ～ 5 を繰り返します。

7. Veritas Cluster Server Manager を起動し、ユーザー名とパスワードを入力してクラスタに接続します。

8. 次の手順で SADatabase エージェントを追加します。
   a. **[File]** » **[Import Types]** をクリックします。
   b. *%VCS_HOME%¥conf¥config¥* に移動して、**[Import]** をクリックします。

### ♦ SADatabase エージェントを使用してフェールオーバーするデータベースの設定

1. 次の手順で、SADatabase をリソースとしてサービスグループに追加します。
   a. **[Edit]** » **[Add]** » **[Resource]** をクリックします。
   b. **[Resource Type]** リストから、**[SADatabase]** をクリックします。

      Windows では、**[Resource Type]** リストに SADatabase が表示されない場合、*SADatabase.xml* ファイルを *%VCS_ROOT%¥cluster manager¥attrpool¥Win2K¥400* に追加して、クラスタサービスを再起動する必要があります。
   c. **[Resource Name]** フィールドに名前を入力します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

d. 以下の各属性について、**[Edit]** 列のボタンをクリックして、次のように属性値を追加します。

- **DatabaseFile**　データベースファイルのロケーション。たとえば、*E:¥demo.db*。
- **DatabaseName**　データベースの名前。
- **ServerName**　データベースサーバーの名前。クラスタ内の各システムには異なるサーバー名を指定できます。属性のスコープは、Global ではなく、Per System にします。
- **UtilDBpwd**　クラスタ内のすべてのシステムで使用されるユーティリティデータベースのパスワード。

e. **[Enabled]** をクリックします。

これにより、リソースの使用準備が整ったことを示します。

f. **[OK]** をクリックします。

2. リソースの依存性が適切に設定されていることを確認します。共有ディスクリソースや IP アドレスリソースなど、SADatabase を起動するために起動/グループ化しておく必要があるリソースが他にも存在します。

3. サービスグループを右クリックして、**[Online]** » [*node-name*] をクリックします。*node-name* は、リソースの実行に使用する、クラスター内のコンピューターの名前です。

これで、サービスグループがオンラインになります。

## SADatabase エージェントのテスト

### ♦ SADatabase エージェントのフェールオーバーのテスト

1. Interactive SQL からデータベースに接続します。次に例を示します。

```
dbisql -c "UID=DBA;PWD=sql;Server=VCS;LINKS=tcpip"
```

2. 次のクエリを実行します。

```
SELECT * FROM Departments;
```

このクエリはエラーなく実行されます。

3. データベースに障害が発生し、第 1 システムノードで実行されているデータベースサーバーがデータベースファイルにアクセスできないとします。これにより、第 2 システムノードで起動されたデータベースサーバーへのデータベースファイルのフェールオーバーが発生します。第 1 ノード上のデータベースファイルに障害を発生させるには、次のようなコマンドを発行します。

```
dbisql -q -c "UID=DBA;PWD=sql;Server=VCS1;DBN=utility_db" STOP DATABASE DEMO ON
VCS1 UNCONDITIONALLY;
```

第 1 コンピューター上のデータベースファイルに障害が発生します。Veritas Cluster Server がこのファイルに障害が発生したことを認識するまでには遅延があります。これは、Veritas

Cluster Server がリソースの状態を監視する間隔がデフォルトでは 60 秒ごとになっているからです (この間隔は、リソース設定で短くできます)。これにより、データベースファイルは第 2 コンピューターにフェールオーバーされ、データベースファイルは第 2 コンピューター上のデータベースサーバーを使用して起動されます。このデータベースサーバーの名前は、元のデータベースサーバーとは名前が異なる場合があります。

たとえば、新しいデータベースサーバーが VCS2 の場合、クライアントでは次のようにして接続文字列にこの新しいデータベースサーバー名を指定する必要があります。

```
"UID=DBA;PWD=sql;Server=VCS2;DBN=DEMO;LINKS=tcpip"
```

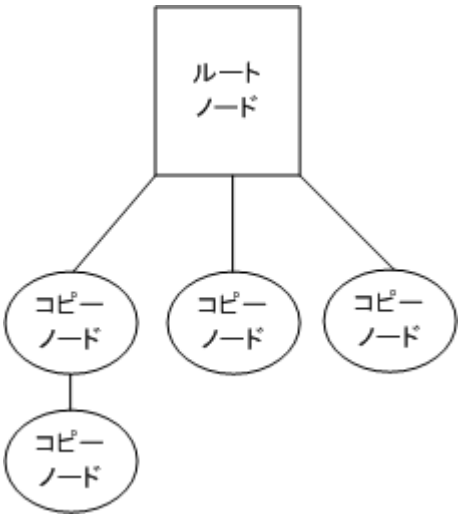
4. Interactive SQL から再接続します。接続もクエリの実行も正常に行われます。

# SQL Anywhere の読み込み専用のスケールアウト

**注意**
読み込み専用のスケールアウトには別のライセンスが必要です。「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

読み込み専用のスケールアウトは、データベースへの読み込み専用アクセスを必要とするレポートなどの操作の負荷を軽減できる設定です。

スケールアウトツリーは、ルートノードとコピーノードで構成されます。スケールアウトシステムの「ルートノード」には、書き込み可能なデータベースの唯一のコピーがあります。ルートノードには、1 つのデータベースまたはデータベースミラーリングシステムを指定できます。ルートノードの下に、コピーノードの 1 つまたは多数の分岐を指定できます。「コピーノード」は、読み込み専用アクセス用にデータベースのコピーを実行するデータベースサーバーであり、コピーノードに他のコピーノードの親を指定できます。




 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

NodeType 接続パラメーターを使用して Kerberos 認証または統合化ログインを使用する接続をリダイレクトしている場合、スケールアウトツリーのすべてのデータベースサーバーを次のように設定する必要があります。

- **統合化ログイン**　スケールアウトツリーのすべてのデータベースサーバーは、同一の Windows ユーザーグループの一部であるか、同一の Windows ユーザーがアクセスすることを許可するように設定する必要があります。

- **Kerberos 認証**　スケールアウトツリーのすべてのデータベースサーバーでは、Kerberos 認証を有効にしておく必要があります。スケールアウトツリーのすべてのサーバーに対する Kerberos サーバープリンシパルは、同一の領域を持つ必要があります。

### スケールアウトの動作

ルートノードは、読み込み要求と書き込み要求の両方を受け入れる唯一のデータベースサーバーです。追加のコピーノードを起動すると、ルートデータベースサーバーは、子として定義されたコピーノードにトランザクションログページを送信します (コピーノードが接続済みでトランザクションログページを受信できる場合)。通常、ページは応答を待たずに送信されますが、処理しきれない量の非同期要求がコピーノードで受信されないようにするため、データベースサーバーから確認が要求されることもあります。

コピーノードは、ページを受信するとそれをディスクに書き込み、子ノード (存在する場合) に送信します。

子ノードが使用できなくなったことを親ノードが検出すると、子へのトランザクションログページの送信が停止します。子が再起動されると、受信されていないトランザクションログページが要求され、親から子へのトランザクションログページの送信が再開します。子はコピーノードのステータス変更を親に通知し、ステータス情報はツリーをたどって最終的にルートデータベースサーバーに到達します。

ルートデータベースサーバーが使用できなくなっても、スケールアウトシステム内のすべての子は引き続き動作しますが、プライマリデータベースサーバーからは更新を受信しなくなります。コピーノード接続では、古くなったデータが取得される可能性があります。ルートデータベースサーバーが再び使用可能になると、子は接続を再確立し、トランザクションログページの受信を再開します。

コピーノードで、互換性がない、トランザクションログが一致しないなどの問題が発生すると、データベースは停止します。問題が発生したデータベースを実行していたデータベースサーバーも、他に実行中のデータベースがないかぎりシャットダウンします。

スケールアウトとデータベースミラーリングを一緒に使用すると、データベースの可用性を保証できます。「データベースミラーリングシステムの読み込み専用スケールアウトシステム」1203 ページを参照してください。

### スケールアウトシステムでのデータベースへの接続

スケールアウトシステムを設定するには、ルートノードとして動作するデータベースサーバーに対して CREATE MIRROR SERVER 文で 2 つの定義 (1 つはパートナーロール用、もう 1 つはプライマリロール用) を作成してください。スケールアウトシステムでは、1 つのパートナーサーバーを定義する必要があります。データベースサーバーを起動するとき、パートナーサーバー定義で指定された名前が接続文字列で使用されます。プライマリサーバーとして定義されたデー

タベースサーバーは、スケールアウトシステムにおけるコピーノードのデフォルトの親になります。プライマリサーバーが定義されていない場合は、システムに追加するコピーノードごとに親を指定してください。

読み込み専用のスケールアウトでは、アプリケーションはルートデータベースサーバーに接続され、アプリケーションの接続文字列をコピーノードのステータスと負荷情報と一緒に使用して、アプリケーションの接続先となるノードをルートデータベースサーバーによって決定することをおすすめします。NodeType=COPY 接続パラメーターを使用して、負荷が最も低いコピーノードにアプリケーションを接続することができます。これを指定すると、ルートデータベースサーバーがクライアントをそのノードにリダイレクトします。「コピーノードへの接続」1201 ページを参照してください。

このような接続と切断をアプリケーションで短時間に何度も行うと、接続がプールされ、どのコピーサーバーを使用するかがルートデータベースサーバーに問われなくなります。この動作により、ルートデータベースサーバーでの負荷は削減されますが、予期する結果にならない可能性があります。接続のたびに接続先となるコピーノードをルートサーバーで確実に判断するため、接続のプールを禁止するようアプリケーションで指定することができます。「接続プールと読み込み専用のスケールアウト」75 ページを参照してください。

**ヒント**
SQL Anywhere モニターを使用してプライマリデータベースをモニタリングすることによって、スケールアウトシステムのデータベースサーバーのステータスを確認できます。「レッスン 3：SQL Anywhere モニターからの読み込み専用スケールアウトシステムのモニタリング」1193 ページと「リソースの追加」1249 ページを参照してください。

また、プライマリデータベースに接続することと、**[正常性と統計情報]** ウィンドウ枠にあるデータベースサーバーのステータスをチェックすることによって、Sybase Central からスケールアウトシステムをモニターすることもできます。「データベースの正常性と統計情報」837 ページと「レッスン 2：Sybase Central からの読み込み専用スケールアウトシステムのモニタリング」1192 ページを参照してください。

# チュートリアル：読み込み専用のスケールアウトシステムの作成とモニタリング

このチュートリアルにあるレッスンを使用すると、コピーノードを自動的に追加するルートデータベースサーバーを設定してモニターする方法を習得します。

**注意**
*%SQLANYSAMP12%¥SQLAnywhere¥DBMirror* には、データベースミラーリングシステムとスケールアウトシステムを組み合わせて使用するサンプルがあります。

## レッスン 1：読み込み専用スケールアウトシステムの作成

このレッスンでは、ルートノード (ルートデータベースサーバー) とコピーノード (データベースの読み込み専用バックアップコピー) を含むスケールアウトシステムを作成します。

**♦ 読み込み専用のスケールアウトシステムを設定するには、次の手順に従います。**

1. *c:¥scaleoutdemo* ディレクトリを作成します。

2. 次のコマンドを実行して、データベース *scaleoutdemo.db* を作成します。これには、次のようにサンプルデータベースのデータが含まれています。

```
newdemo c:¥scaleoutdemo¥scaleoutdemo.db
```

3. スケールアウトシステムのルートデータベースサーバーを起動します。

```
dbsrv12 -n scaleout_root_demo -su sql -x TCPIP(port=6871) "c:¥scaleoutdemo¥scaleoutdemo.db" -
xp on
```

-su オプションを使用して、ユーティリティデータベースのパスワードを指定することをおすすめします。次に、必要があれば、データベースサーバーをシャットダウンするユーティリティデータベースに接続することもできます。 「-su dbeng12/dbsrv12 サーバーオプション」276 ページを参照してください。

4. Interactive SQL からルートデータベースに接続します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=scaleout_root_demo;DBN=scaleoutdemo"
```

5. スケールアウトシステムにルートデータベースサーバーを定義します。

```
BEGIN
   EXECUTE IMMEDIATE
   'CREATE MIRROR SERVER "scaleout_primary_demo"
   AS PRIMARY
      connection_string = "SERVER=scaleout_primary_demo;HOST='
      || PROPERTY( 'MachineName' ) || ':6871'";
END
```

6. スケールアウトシステムでルートデータベースサーバーをパートナーとして定義します。パートナーサーバー名は、データベースサーバーの起動コマンドで使用したデータベースサーバー名と一致している必要があります。

```
BEGIN
   EXECUTE IMMEDIATE
   'CREATE MIRROR SERVER "scaleout_root_demo"
      AS PARTNER
      connection_string = "SERVER=scaleout_root_demo;HOST='
      || PROPERTY( 'MachineName' ) || ':6871'";
END
```

7. スケールアウトシステムのルートサーバーにオプションを設定します。

```
SET MIRROR OPTION auto_add_server='scaleout_root_demo';
SET MIRROR OPTION child_creation='automatic';
SET MIRROR OPTION authentication_string='abc';
SET MIRROR OPTION auto_add_fan_out='10';
```

8. データベースのバックアップコピーを作成し、ディレクトリ *c:¥scaleoutdemo¥copynode* に格納します。

```
BACKUP DATABASE DIRECTORY 'c:¥¥scaleoutdemo¥¥copynode';
```

9. データベースのバックアップコピーを scaleout_root_demo データベースサーバーの子 (コピーノード) として起動します。

```
dbsrv12 -n scaleout_child_demo -su sql -x TCPIP(port=6873) "c:¥scaleoutdemo¥copynode
¥scaleoutdemo.db" -xp on
```

10. Interactive SQL から子ノードに接続します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=scaleout_child_demo"
```

子ノードが親に接続すると、子ノードがデータベースの読み込み専用コピーであることを示す警告が表示されます。これで、子ノードに接続し、データベースのコピーに対してクエリを実行できるようになります。

11. 次のクエリを実行すると、スケールアウトシステムでミラーサーバーを表示できます。

```
SELECT * FROM SYSMIRRORSERVER;
```

12. 「レッスン 2：Sybase Central からの読み込み専用スケールアウトシステムのモニタリング」1192 ページに進みます。

## レッスン 2 : Sybase Central からの読み込み専用スケールアウトシステムのモニタリング

このレッスンでは、Sybase Central を起動し、データベース *scaleoutdemo.db* に接続して、Sybase Central から読み込み専用スケールアウトシステムをモニターする方法を習得します。

**♦ Sybase Central から読み込み専用スケールアウトシステムをモニターするには、次の手順に従います。**

1. Sybase Central を起動します。**[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[Sybase Central]** をクリックします。
2. **[接続]** » **[SQL Anywhere 12 に接続]** をクリックします。
3. **[接続]** ウィンドウを使用して、データベース *scaleoutdemo.db* に接続します。
   a. **[ユーザー ID]** フィールドに **DBA** と入力します。
   b. **[パスワード]** フィールドに **sql** と入力します。
   c. **[サーバー名]** フィールドに **scaleout_root_demo** と入力します。
   d. **[接続]** をクリックします。
4. メインウィンドウにある **[概要]** タブをクリックして、データベースの *scaleoutdemo.db* をモニターします。
5. **[データベースのミラーリングおよびスケールアウト]** ドロップダウンリストをクリックして、スケールアウトシステムのプライマリサーバー名を、スケールアウトシステムのコピーノードに関する情報と併せて表示します。
6. Sybase Central をシャットダウンします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

a. [F12] キーを押してデータベースから Sybase Central を切断します。

b. **[ファイル]** » **[終了]** をクリックします。

7. 「レッスン 3：SQL Anywhere モニターからの読み込み専用スケールアウトシステムのモニタリング」1193 ページに進みます。

## レッスン 3：SQL Anywhere モニターからの読み込み専用スケールアウトシステムのモニタリング

このレッスンでは、読み込み専用スケールアウトシステムでルートノードである SQL Anywhere データベースを起動し、このデータベースを、SQL Anywhere モニターからモニタリングされるリソースとして追加します。

♦ **データベースリソースをモニターに追加するには、次の手順に従います。**

1. SQL Anywhere モニターを起動します。

   - **モニター Developer Edition を起動するには、次の手順に従います (Windows の場合)。** **[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[SQL Anywhere モニター]** をクリックします。
   - **モニター Developer Edition を起動するには、次の手順に従います (Linux の場合)。** モニターのインストールディレクトリの *bin32* または *bin64* ディレクトリから *samonitor.sh* スクリプトを実行します。

     ```
     samonitor.sh launch
     ```

2. SQL Anywhere モニターにログインします。

   - **デフォルトの管理者ユーザーとしてモニターにログインします。** **[ユーザー名]** フィールドに **admin** と入力し、**[パスワード]** フィールドに **admin** と入力します。

3. 新しいデータベースリソースを追加します。

   読み込み専用スケールアウトシステムとその他のデータベースサーバーで、ルートノード用のリソース設定には違いはありません。リソースの設定時や移行時に、モニターはリソースが読み込み専用スケールアウトシステムのルートノードであることを検出します。

   a. 左のサイドバーにある **[ツール]** » **[管理]** をクリックします。

   b. **[リソース]** をクリックして、**[追加]** をクリックします。

   c. **[SQL Anywhere サーバー]** をクリックして **[次へ]** をクリックします。

   d. **[名前]** フィールドに *scaleoutdemo* と入力し、**[次へ]** をクリックします。

   e. **[ホスト]** フィールドに **localhost** と入力し、**[サーバー]** フィールドに **scaleout_root_demo** と入力します。

   f. **[作成]** をクリックします。

4. 必要な認証情報が要求されたら、**[DBA ユーザー ID]** フィールドに **DBA** と入力し、**[パスワード]** フィールドに **sql** と入力します。**[OK]** をクリックします。

データベースをモニタリング対象のリソースとして追加する場合は、データベースの DBA ユーザー ID とパスワードを指定する必要があります。これらのクレデンシャルは次の操作を行う場合に使用されます。

- データベースに接続します。
- **sa_monitor_user** という名前の新しいユーザーを作成します。モニターは、sa_monitor_user を使用してデータベースに接続し、モニタリングします。
- データベースをモニタリングするために **sa_monitor_user** が必要とするデータベースオブジェクトをインストールします。インストールされるオブジェクトのリストについては、「モニターがインストールしたデータベースオブジェクトのリスト」1311 ページを参照してください。
- モニターから DBA クレデンシャルを破棄します。**sa_monitor_user** はモニタリングされるデータベースに追加されるので、そのデータベース外部のどこかに DBA クレデンシャルを格納する必要はありません。

リソースが追加され、モニタリングが自動的に開始されます。

5. **[閉じる]** をクリックします。
6. **[閉じる]** をクリックします。
7. **[ダッシュボード]** » **[概要]** をクリックします。

   **[リソースリスト]** に scaleoutdemo リソースが表示されます。
8. **[scaleoutdemo]** をクリックすると、scaleoutdemo ダッシュボードが開き、収集されたメトリックが表示されます。

### ♦ [SQL Anywhere スケールアウトトポロジ] ウィジェットの追加

1. scaleoutdemo ダッシュボードの右上端で、**[カスタマイズ]** » **[ウィジェットの追加]** » **[SQL Anywhere スケールアウトトポロジ]** をクリックします。
2. **[次へ]** をクリックします。
3. **[このウィジェットの名前を指定してください。]** フィールドに **scaleoutwidget** と入力します。
4. **[どのようなリソースに関心がありますか ?]** フィールドで **scaleoutdemo** を選択します。
5. **[作成]** をクリックします。

ダッシュボードに新しいウィジェットが表示されます。

| サーバー名 | タイプ | 状態 | 接続 | 最終更新 |
|---|---|---|---|---|
| mirror_server2 | partner - primary | connected | 1 | 2011/04/05 13:33 |
| mirror_server1 | partner - mirror | disconnected | 0 | 2011/04/05 13:22 |
| demo_arbiter | arbiter | connected | | |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

トポロジウィジェットでは、次のカラムとその使用できる値が表示されます。

- **[サーバー名]**　スケールアウトシステムのノード名。
- **[タイプ]**　サーバーのロール。タイプは、読み込み専用スケールアウトシステムが作成されたときにサーバーが定義された方法によって、**[partner - primary]** または **[コピー]** のいずれかになります。
- **[状態]**　ノードの接続状況。**[接続しました]** または **[切断しました]** のいずれか 1 つになります。
- **[接続]**　このノードへの現在の接続数。
- **[最終更新]**　サーバーがステータスを最後に更新した時刻。

「レッスン 4：モニターからのログアウトと、読み込み専用スケールアウトシステムのシャットダウン」1195 ページに進みます。

## レッスン 4：モニターからのログアウトと、読み込み専用スケールアウトシステムのシャットダウン

このレッスンでは、SQL Anywhere モニターからログアウトして、scaleout_child_demo および scaleout_root_demo database サーバーをシャットダウンします。

**♦ モニターをログアウトして、読み込み専用システムをシャットダウンするには、次の手順に従います。**

1. モニターからログアウトします。

   **[ログアウト]** をクリックします。

2. SQL Anywhere モニターを表示していたブラウザーウィンドウを閉じます。

3. モニターを終了し、モニタリングを停止します。

   - Windows では、システムトレイで [SQL Anywhere モニター] アイコンを右クリックし、**[SQL Anywhere モニターの終了]** をクリックします。
   - Linux では、モニターのインストールディレクトリの *bin32* または *bin64* ディレクトリから *samonitor.sh* スクリプトを実行します。

     ```
     samonitor.sh stop
     ```

   モニターは、すべてのリソースについてメトリックの収集を停止します。

4. SQL Anywhere データベース *scaleoutdemo.db* をシャットダウンします。

   a. システムトレイで scaleout_child_demo データベースサーバーのネットワークサーバーアイコンをダブルクリックします。

   b. データベースサーバーメッセージウィンドウで **[シャットダウン]** をクリックします。

c. **[はい]** をクリックします。

d. scaleout_root_demo データベースサーバーに対してこの手順を繰り返します。

## 読み込み専用のスケールアウトの設定

次の手順を実行して、スケールアウトシステムを設定してください。

1. **-xp on** データベースオプションを使用して、データベースサーバーを起動します。このデータベースサーバーは、スケールアウトシステムのルートノードです。

2. ルートデータベースにスケールアウトのオブジェクト定義を追加します。サーバーを定義し、サーバーのオプションを設定してください。

3. ルートデータベースのバックアップコピーを作成します。

4. コピーノード (データベースのバックアップコピー) を起動します。

## ルートノードの設定

コピーするデータベースを使用して、読み込み専用スケールアウトシステムのルートノードを作成します。

**前提条件**

なし

**内容と備考**

ルートノードを設定する場合、ルートデータベースサーバーをパートナーサーバーとして定義します。また、プライマリサーバーも定義できます。

**♦ ルートノードの設定**

1. 読み込み専用コピーにするデータベースが稼働しているデータベースサーバーを起動します。データベースサーバーを起動する場合は、**-xp on** データベースオプションを指定してください。

2. CREATE MIRROR SERVER 文を使用して、ルートデータベースでスケールアウトオブジェクトを定義します。

   ルートデータベースサーバーに対して、CREATE MIRROR SERVER...AS PARTNER と CREATE MIRROR SERVER...AS PRIMARY の 2 つの定義を作成することをおすすめします。AS PARTNER 句を指定する文でデータベースサーバーに付ける名前は、データベースサーバーの起動コマンドとクライアント接続文字列で使用する名前です。AS PRIMARY 句を指定する文でデータベースサーバーに付ける名前は、スケールアウトシステムに追加するコピーノードのデフォルトの親となるデータベースサーバーの名前です。プライマリサーバーを定義しないと、コピーノードを作成するときに親サーバー名の指定が必要となるため、プライマリサーバーを定義することをおすすめします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

3. SET MIRROR OPTION 文を使用して、データベースにスケールアウトオプションを設定します。

**結果**

これで、ルートノードが使用可能になりました。

**次の手順**

読み込み専用スケールアウトシステムに子ノードを追加するのに使用するルートデータベースのバックアップコピーを作成することができます。

**参照**


- 「-xp dbsrv12 データベースオプション」318 ページ
- 「CREATE MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』
- 「SET MIRROR OPTION 文」『SQL Anywhere サーバー SQL リファレンス』
- 「子コピーノードを追加する方法」1197 ページ


# 子コピーノードを追加する方法

SET MIRROR OPTION 文の child_creation オプションは、読み込み専用のスケールアウトシステムに子ノードを追加する方法を制御します。次の値がサポートされます。

- **Automatic**　ルートデータベースサーバーでは、コピーノードが起動したときにそれを認証するか、またはコピーノードが不明の場合は新しく作成します。コピーノードを手動で作成しなくても、不明なコピーノードの定義がルートサーバーで作成されるため、この設定をおすすめします。

- **Off**　ルートデータベースサーバーに接続し、CREATE MIRROR SERVER 文を実行して新しいコピーノードを作成する必要があります。

- **Manual**　コピーノードに接続し、そのデータベースサーバーに対して CREATE MIRROR SERVER 文を実行することによって、コピーノードをツリーに追加できます。この文の実行には、DBA 権限が必要です。コピーサーバーは、新しいコピーノードの定義要求をルートデータベースサーバーに送信します。コピーノードが定義されると、ルートデータベースサーバーにより新しいコピーノードからログページを要求できるようになります。

データベースには、システム内でプライマリデータベースサーバーに関連付けられている接続文字列が格納されます。前に定義されていない新しいコピーノードを起動すると、この接続文字列を使用してルートデータベースサーバーに接続されます。ルートデータベースサーバーでは、データベースに格納されている authentication_string オプションの値を使用して、コピーノードが認証されます。

コピーノードはミラー接続要求の一部として、コピーデータベースサーバーの名前と、その IP アドレスとポートを含む文字列をルートデータベースサーバーに送信します。コピーノードが認証されると、それが認識されるかどうかがルートデータベースサーバーで判断されます。コピーノードが不明の場合は、ルートデータベースサーバーで CREATE MIRROR SERVER 文が実行され、新しいコピーとその接続文字列が定義されます。コピーが認識されると、ルートデータ

ベースサーバーはコピーへの接続を確立できます。次に、コピーにまだ存在しないすべてのトランザクションログページがコピーから要求され、それがすべて取得された時点で、ルートデータベースサーバーは新しいトランザクションログページをコピーノードに送信し始めます。

すべてのコピーノードサーバーの名前がユニークになるように指定します。

**参照**

- 「SET MIRROR OPTION 文」『SQL Anywhere サーバー SQL リファレンス』
- 「CREATE MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』

## コピーノードの追加

コピーノードの定義がデータベースに保存されます。スケールアウトシステムのコピーノードを事前に定義するか、コピーノードの接続時にルートデータベースサーバーによって定義することができます。

### 前提条件

なし

### 内容と備考

ルートデータベースサーバーによってコピーノードを定義すると、定義時のエラーの発生率が抑えられるため、この方法をおすすめします。

### ♦ コピーノードの自動定義

1. ルートデータベースのバックアップコピーを作成します。
2. データベースのコピーをコピーノードとして起動します。コピーノードを実行するデータベースサーバーを起動するときは、**-xp on** データベースオプションを指定する必要があります。

### 結果

データベースは、コピーノードとして起動されます。

### 次の手順

なし

**参照**

- 「-xp dbsrv12 データベースオプション」318 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

# コピーノードの親の割り当て

コピーノードには親が必要です。ルートデータベースサーバーまたは別のコピーノードを親として指定できます。親が自動的に割り当てられるように、読み込み専用のスケールアウトシステムを設定できます。また、スケールアウト階層内でのノードの位置をいつでも変更できます。

## コピーノードへの親の自動割り当て

SET MIRROR OPTION 文では、トランザクションログページの分散処理をノード間でバランスが取れるようにするため、新しいコピーノードをツリー内の親に割り当てるときに使用できる 2 つのオプションがサポートされています。

- **auto_add_server**　自動割り当てツリーの最上位ノードとして動作するデータベースサーバーの名前を指定します。
- **auto_add_fan_out**　ツリー内の各分岐の最大コピーノード数を指定します。指定可能な最小値は 2 で、デフォルトは 10 です。次の操作の結果作成される新しいノードには、親が自動的に割り当てられます。
  - child_creation オプションが Automatic に設定されている場合のプライマリサーバーへの接続
  - child_creation が手動に設定されている場合の、コピーノードでの CREATE MIRROR SERVER...USING AUTO PARENT の実行
  - ルートデータベースサーバーでの CREATE MIRROR SERVER...USING AUTO PARENT の実行

auto_add_server オプションが指定されている場合は、そのオプションで指定されたデータベースサーバーが最上位のノードとして使用されます。add_auto_server オプションが指定されていない場合は、自動割り当てを使用してルートデータベースサーバーがすべてのコピーノードの親として使用され、auto_fan_out オプションは無視されます。

ルートデータベースサーバーは、ツリー内のどのノードを新しいコピーノードの親として割り当てるかを、定期的なステータス更新によってルートデータベースサーバーに通知される各ノードの子の数とそのステータスに基づいて決定します。コピーノードでは、ルートデータベースサーバーから受け取ったトランザクションログ操作を適用するときに、新しい親の割り当てがわかります。したがって、コピーノードは、トランザクションログ操作を受け取った時点で親の接続を変更します。

**参照**

- 「SET MIRROR OPTION 文」『SQL Anywhere サーバー SQL リファレンス』
- 「CREATE MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』

## ノードの位置

コピーノードがいったん実行中になると、ルートデータベースサーバーで ALTER MIRROR SERVER 文を使用して、新しい親をコピーノードに割り当てることができます。

```
ALTER MIRROR SERVER "copy-server-name"
 FROM SERVER "new-parent-name";
```

この文は、ルートデータベースサーバー上のトランザクションログに記録されます。変更内容がコピーノードに適用されると、コピーノードで定義の変更が認識され、文で指定された新しい親に接続されます。

**参照**

- 「ALTER MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』

## コピーノードの親の判断

コピーノードは、ISYSMIRRORSERVER システムテーブルでコピーノードのミラーサーバー定義と親の値を確認することによって、接続先となるサーバーを特定します。

- NULL は、コピーノードの親がルートノードであることを示します。
- 親カラムに ID が表示されている場合は、その ID を持つサーバーがコピーノードの親になります。

コピーノードで親が特定されると、データベースに格納されている connection_string オプションを使用して、正しい親に接続されます。

### データベースミラーリングシステムでのコピーノードの親の特定

スケールアウトとデータベースミラーリングシステムを一緒に使用している場合は、コピーノードのパートナーがプライマリサーバーまたはミラーサーバーとして現在動作しているサーバーとなるようにコピーノードを定義できます。この設定では、ISYSMIRRORSERVER で、1 つのサーバーがタイプ PRIMARY、もう 1 つのサーバーがタイプ MIRROR として定義されます。どちらのサーバーも、コピーノードの親として選択できます。

**参照**

- 「SYSMIRRORSERVER システムビュー」『SQL Anywhere サーバー SQL リファレンス』
- 「SET MIRROR OPTION 文」『SQL Anywhere サーバー SQL リファレンス』

# 親接続の消失の処理

コピーノードの親サーバーが使用できなくなっても、コピーノードのデータベースは引き続き使用できます。コピーノードは、max_retry_connect_time ミラーオプションで定義された期間 (デフォルトでは 120 秒) 中は引き続き親に接続しようとします。指定期間内に接続を確立できない場合、代替の親 (CREATE MIRROR SERVER STATEMENT の OR SERVER 句を使用して定義) が定義されていれば、その親に接続しようとします。代替の親が定義されている場合は、さらに

max_retry_connect_time 秒間、接続を試みます。接続を確立できないと、コピーノードはルートデータベースサーバーへの接続を試み、以前の親をコピーノード自身と置き換えコピーノードの兄弟を子として割り当てるよう要求します。この動作により、auto_add_fan_out 設定で指定された個数を超える子がコピーノードに存在するという状態になることがあります。

親接続が消失してから max_disconnected_time 秒 (デフォルトでは無制限) 以内に、コピーサーバーからルートデータベースサーバーへの接続を確立できないと、データベースは停止します。

コピーサーバーから別のデータベースサーバーへの接続を確立しようとする間は、元の親への接続を引き続き試みます。この接続に成功すると、元の親データベースサーバーからログページの取得が再開します。

読み込み専用ノードが最初に起動された場合は、ツリー内の他のノード (親も含む) が使用できなくても、読み込み専用の接続に関してはデータベースが使用可能になります。

### スケールアウトシステムでのサーバー状態の特定

次の文を使用して、コピーノードの親への接続状態を判断できます。

```
SELECT DB_EXTENDED_PROPERTY('MirrorServerState',server-name );
```

次の文を使用して、任意のノードにおけるトランザクションログの処理量を把握できます。

```
SELECT DB_PROPERTY('CurrentRedoPos');
```

子ノードの CurrentRedoPos プロパティの値を、プライマリサーバーの同じプロパティと比較して、コピーサーバーがどの程度最新の状態であるかを判断できます。

sa_mirror_server_status システムプロシージャーから、プロシージャーを呼び出したサーバーの下にあるすべてのサーバーの接続状況が返されます。

### 参照

- 「sa_mirror_server_status システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「SET MIRROR OPTION 文」『SQL Anywhere サーバー SQL リファレンス』
- 「CREATE MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』

## コピーノードへの接続

プライマリサーバーに接続し、接続文字列の一部として NodeType 接続パラメーターを指定すると、コピーノードに接続できます。NodeType 接続パラメーターには、次の値を指定できます。

- **COPY**　NodeType 接続パラメーターが COPY に設定されている場合、データベースサーバーはその分岐内のコピーノード (ルートノードでない場合はサーバー自体も) を検査し、負荷が最も軽いコピーノードを選択します。データベースサーバーがサーバー自体を選択しない場合、選択したデータベースサーバーにクライアントをリダイレクトします。

- **DIRECT**　これがデフォルト設定です。NodeType 接続パラメーターが DIRECT に設定されている場合、データベースサーバーは、負荷分散またはリダイレクションを行わずに接続を受け入れます。

- **PRIMARY**　プライマリサーバーに接続したときに、NodeType 接続パラメーターが PRIMARY に設定されている場合、接続は受け入れられます。ミラーサーバーまたはコピーノードに接続した場合、データベースサーバーは接続をプライマリサーバーにリダイレクトします。

接続されるコピーノードがルートデータベースサーバーで判断されると、そのノードに接続が自動的に (必要に応じて) リダイレクトされます。

**参照**

- 「NodeType (NODE) 接続パラメーター」357 ページ

# 読み込み専用スケールアウトシステムからのサーバーの削除

スケールアウトシステムを設定するとき、ルートデータベースサーバーに 2 つの定義 (CREATE MIRROR SERVER...AS PARTNER と CREATE MIRROR SERVER...AS PRIMARY) がある可能性があります。ルートサーバーの削除理由によっては、ロール定義とパートナーサーバー定義の両方をデータベースから削除する必要がある場合があります。

### 前提条件

この作業を実行するための前提条件は、ありません。

### 内容と備考

次のとおりです。

**♦ ミラーサーバーの削除 (SQL の場合)**

> **注意**
> 同じサーバー名を保持して設定内容を変更する場合は、CREATE OR REPLACE MIRROR SERVER 文または ALTER MIRROR SERVER 文を使用できます。

1. DBA 権限のあるユーザーとして、データベースに接続します。
2. DROP MIRROR SERVER 文を実行してミラーサーバーを削除します。次に例を示します。

### 結果

ミラーサーバーが読み込み専用スケールアウトシステムから削除され、その定義もデータベースから削除されます。

### 次の手順

なし

### 例

次の文により、mymirror1 という名前のミラーサーバーが削除されます。

```
DROP MIRROR SERVER mymirror1;
```

**参照**

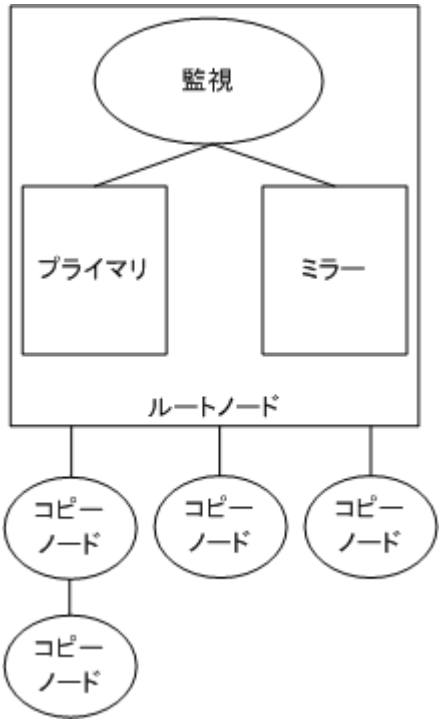
- 「DROP MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』
- 「CREATE MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』
- 「ALTER MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』

# データベースミラーリングシステムの読み込み専用スケールアウトシステム

Web アプリケーションをホストするために高可用性かつ読み込み専用のスケールアウトを使用する方法の詳細については、「SQL Anywhere Web サービスの高可用性とスケールアウトソリューション」1205 ページを参照してください。

データベースミラーリングシステムでは、読み込み専用の接続を使用して、ミラーサーバー上で実行されているデータベースにアクセスできます。この機能は、このデータベースへの読み込み専用アクセスを必要とするレポートなどの操作の負荷を軽減したい場合に役立ちます。ミラーサーバーとコピーノードの唯一の違いは、コピーノードがプライマリサーバーまたはミラーサーバーとしてデータベースミラーリングシステムに加わることができない点です。コピーノードは、それがコピーするデータベースの監視サーバーとして動作することができます。

読み込み専用のスケールアウトでデータベースミラーリングを使用することで、ルートノードの可用性を保証できます。データベースミラーリングでスケールアウトを使用する場合、スケールアウトシステムのルートノードは、単一のデータベースサーバーで構成されるのではなく、プライマリサーバー、ミラーサーバー、監視サーバーで論理的に構成されます。




ミラーリングシステムでデータベースサーバーを起動すると、参加しているミラーサーバーに他のデータベースサーバーからアクセスして、データベースの読み込み専用コピーを管理できます。必要に応じて、コピーノードの分岐をいくつでもミラーサーバーに追加できます。

### コピーノードとプライマリサーバー／ミラーサーバーの違い

データベースミラーリングシステムでコピーノードを使用する場合、現在のプライマリサーバー、現在のミラーサーバー、または別のコピーノードの子としてコピーノードを指定できます。コピーノードは、その親からトランザクションログページを取得します。ミラーノードと異なり、コピーノードにはステータス情報ファイルはありません。これは、どのデータベースサーバーにデータベースの所有権を与えるかについて、コピーノードのステータスが影響しないためです。コピーノードは、コピーしているデータベースの監視サーバーとしても動作することができます。

### 参照


- 「データベースミラーリング」1137 ページ
- 「ミラーサーバーで実行されているデータベースへの読み込み専用アクセスの設定」1165 ページ
- 「監視サーバーとしてコピーノードを使用する」1167 ページ
- 「ISYSMIRRORSERVER システムテーブル」『SQL Anywhere サーバー SQL リファレンス』
- 「CREATE MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』

# SQL Anywhere Web サービスの高可用性とスケールアウトソリューション

次の SQL Anywhere コンポーネントは、Web アプリケーションをホストするための高可用性やスケールアウトソリューションの設定に必要です。

- **Relay Server (RS)**　Relay Server は、HTTP 要求のエントリポイント (通常は Internet から) であり、仮想ホストと負荷分散機能を提供します。Relay Server は、「サーバーファーム」という仮想エンティティをターゲットとする HTTP 要求を受け入れ、ルート指定します。各サーバーファームは、1 つ以上の SQL Anywhere データベース／Web サーバーで構成できます。Relay Server は、処理用に適したバックエンドサーバーにすべての HTTP 要求をリレーする HTTP/WEB アクセスポイントです。「Relay Server の概要」『Relay Server』を参照してください。

- **Outbound Enabler (OE)**　Outbound Enabler は、Relay Server とバックエンドデータベースサーバー間のブリッジの役割を果たします。サーバーファーム内にあるバックエンドサーバーごとに、1 つの Outbound Enabler プロセスがあります。その目的は、指定されたバックエンドサーバーが稼働していることを確認し、Relay Server と間の通信チャネルを確立して要求をバックエンドサーバーに転送することにあります。Outbound Enabler は Relay Server への接続を開始するため、ファイアウォール経由で LAN から DMZ にある Relay Server にアウトバウンド接続を簡単に行うことができます。「Outbound Enabler」『Relay Server』を参照してください。

- **SQL Anywhere (バックエンド) サーバー**　SQL Anywhere (バックエンド) サーバーは、完全に SQL で作成された Web アプリケーションをホストできる、高度なデータベースであり Web サーバーです。SQL Anywhere サーバーは、高可用性とスケールアウトソリューションを必要とする、負荷の高い環境を緩和するために配備できます。高可用性により、サービスの中断から保護するフォールトトレランス機能が提供され、スケールアウトにより、負荷のピーク時にスループットを向上するための負荷分散が行われます。

## 高可用性

高可用性を提供するには、3 つの SQL Anywhere サーバー (1 つの「監視」サーバーと 2 つの「パートナー」サーバー) が必要です。1 つのパートナーサーバーには「プライマリ」ロールが割り当てられ、もう 1 つには「ミラー」ロールが割り当てられます。グローバルテーブルへの挿入などの書き込み操作はプライマリサーバーでのみ許可されますが、選択などの読み込み操作はどちらのタイプのサーバーでも許可されます。プライマリサーバーは、更新を受信するたびにミラーサーバーを同期します。プライマリサーバーで障害が発生した場合は、監視サーバーによって、ミラーサーバーがプライマリサーバーとなるようにパートナーのロールが切り替えられます。

Relay Server 経由ですべての HTTP 要求をルート指定することにより、Web サービス環境内で高可用性を活用できます。Relay Server では、HTTP 要求の URL を処理して、ターゲットサーバーファームが抽出されます。わかりやすくするため、HTTP 要求を readFarm または writeFarm のどちらかにルート指定する Relay Server 設定について考えてみます。プライマリバックエンドサーバーでは読み込み要求と書き込み要求の両方を処理できますが、サーバー障害が発生したときに読み込み機能を処理する他の読み込み専用コピーノードがある場合には書き込み要求のみを処理するよう設定し、そうでない場合には読み込み要求と書き込み要求の両方を処理するように設定できます。ただし、ミラーサーバーで読み込み要求を処理して読み込み負荷を分散した

り、ミラーサーバーでのハードウェア使用を削減するために要求をいっさい処理しないように設定したりできます。クライアントアプリケーションでは、読み込み要求と書き込み要求用にアクセスする物理サーバーを認識する必要はなく、要求を readFarm または writeFarm のどちらかにダイレクトするように URL を構成するにすぎません。

Relay Server に加え、パートナーである各バックエンドサーバーには、2 つの Outbound Enabler が必要です (バックエンドサーバーが参加している各ファームに 1 つずつ)。2 つの Outbound Enabler は、プライマリパートナーへの通信チャネルを制御し、ミラーパートナーとの通信を確立します。この設定では、プライマリサーバーは常に 1 つしか存在しません。プライマリサーバーは、writeFarm に割り当てられた Outbound Enabler をアクティブにし、コピーノード以外のメンバーが readFarm に存在する場合は、オプションとして readFarm に関連付けられている Outbound Enabler を非アクティブにします。ミラーサーバーは、writeFarm に関連付けられている Outbound Enabler を非アクティブにし、オプションとして readFarm に関連付けられている Outbound Enabler をアクティブにします。

ハードウェアやソフトウェアの障害によりプライマリサーバーが使用できなくなった場合、ミラーサーバーが監視サーバーとネゴシエートしてデータベースのロールを引き受け、これにより writeFarm のメンバーとなります。また、他の読み込み専用コピーノードがシステムに存在しない場合は、readFarm のメンバーにもなります。障害が発生すると、元のプライマリサーバーが writeFarm と readFarm のメンバーである場合は、その両方への参加が突然解消されます。元のプライマリサーバーは、障害の原因が究明するまで回復しないことがあります。回復した場合にはミラーロールが想定され、オプションとして readFarm に加わります。サーバーの突然の障害により、そのサーバーで処理されていた要求が失敗することがあります。ミラーサーバーからの writeFarm の回復の遅延は、SQL Anywhere ミラーのロール切り替えの遅延と、ミラーの writeFarm Outbound Enabler で指定されたステータスポーリング間隔によって異なります。このステータスポーリング間隔は、明示的に指定されていない場合は設定できます。デフォルトは 5 秒です。

### スケールアウト

スケールアウト機能を提供するには、最低でも 2 つの SQL Anywhere サーバー (ルートノードとコピーノード) が必要です。コピーノードは、ミラーサーバーと同様に読み込み専用ですが、プライマリロールが想定されることがないという点でミラーサーバーとは異なります。コピーノードは、読み込み要求のサービスに限定されています。高可用性の項で説明したように、プライマリサーバーは writeFarm に関連付けられ、コピーノードは readFarm メンバーとしてミラーサーバーに置き換わります。1 つのルートノードと 1 つのコピーノードの場合は、合計 2 つの Outbound Enabler のみが必要です (各サーバーに 1 つずつ)。すべてのコピーノードが停止しても環境が完全に機能できるようにするため、オプションとしてプライマリサーバーを readFarm に追加できます。ルートノードに関連付けられている Outbound Enabler は、writeFarm への要求を受信するように設定されます。一方、コピーノードの Outbound Enabler は、readFarm への要求を受け入れるように設定されます。ルートノードは、書き込み更新を受信した時点でコピーノードを継続的に更新します。コピーノードを readFarm に追加することで、設定を拡張できます。readFarm に対して 10 個のコピーノードを実行すると、負荷分散が行われ、readFarm 内で発生するサーバー障害が緩和されます。

### 高可用性とスケールアウト

上記の設定シナリオを組み合わせて、高可用性とスケールアウトの両方を提供することができます。このような設定では、書き込み可能 (プライマリ／ミラー) と読み込み可能 (ミラーとコピー) のバックエンドサーバーに対してフォールトトレランスが提供されます。また、コピー

ノードをさらに追加することで、HTTP 読み込み専用要求に対する潜在的なスループットが高まります。この設定では、プライマリサーバーとミラーサーバーがルートノードを構成し、プライマリノード、ミラーノード、コピーノードのすべてに Outbound Enabler がそれぞれ 1 つずつ関連付けられて readFarm に加わります。プライマリサーバーとミラーサーバーには、writeFarm に加わるための追加の Outbound Enabler があります。オプションとして、プライマリサーバーとミラーサーバーを readFarm から削除できます。

### OE 設定

この説明では、次の項目を識別するために Outbound Enabler を設定してください。

●関連付けられたサーバーファーム

●Relay Server 接続パラメーター

●SQL Anywhere (バックエンドサーバー) 接続パラメーター

次のコマンドで、-cr オプションで指定されたパラメーターに基づいて Relay Server への接続を設定する Outbound Enabler を示します。Relay Server がアクティブの場合は、readFarm に対して行われた要求を、-cs オプションに基づいて設定されたバックエンドサーバーに転送します。

```
rsoe.exe -f readFarm -id saHost1 -cr url_suffix="/ias_relay_server/server/
rs_server.dll";host=rs_server;port=80 -cs host=sahost1;port=8080;status_url=/satest/oe_read_status?
ro=1
```

上記の例に基づいて、http://rs_server/ias_relay_server/client/rs_client.dll/readFarm/service_path[?service_query] の形式の要求 URL は、

rs_server サーバーでホストされている Relay Server に要求をダイレクトします。Relay Server は、URL からドメインとファームの各コンポーネントを削除し、ファームに関連付けられた Outbound Enabler に要求を RS-OE チャネル経由でリレーします。次に、Outbound Enabler は要求をバックエンドサーバーに転送します。

バックエンドサーバーで要求の処理準備が整ったことが Outbound Enabler で検出された場合にのみ、Relay Server へのチャネルが保持されます。検出するには、設定可能な間隔でサーバーをポーリングし、-cs オプションで指定された HTTP 要求を送信します。status_url オプションは、ping サービス用の特定の URL パスを指定します。上記の設定例では、HTTP 要求が Outbound Enabler から http://sahost1:8080/satest/oe_read_status?ro=1 に送信されます。サーバーから **AVAILABLE=TRUE | ON | 1** と応答されると、Outbound Enabler は、rs_server でホストされる Relay Server との通信チャネルを保持または開始します。

### SQL Anywhere の設定

SQL Anywhere を Web サーバーとして設定および起動するには、「HTTP Web サーバーとしての SQL Anywhere」『SQL Anywhere サーバー プログラミング』を参照してください。

Outbound Enabler の設定の項で説明している RSOE (Relay Server Outbound Enabler) 起動例を活用して、oe_read_status サービスを次のように作成できます。

```
// READ-ONLY SA-OE ping service
//  optionally includes primary as a READ node.
call sa_make_object( 'service', 'oe_read_status' );
alter service oe_read_status
```

```
    type 'raw'
    authorization off
    secure off
    user DBA
    as call sp_oe_read_status(:ro);

// if ro is 1 then only read-only servers are activated: mirror or copy
// if ro is 0 then include all servers: primary, mirror, copy...
create or replace procedure sp_oe_read_status( ro int )
begin
    declare readonly long varchar;
    declare res long varchar;

    set res = 'AVAILABLE=';

    set ro = isnull(ro, 0);
    if ro = 1 then
        select db_property('ReadOnly') into readonly;
         if readonly = 'On' then
             set res = res || 'TRUE';
         else
             set res = res || 'FALSE';
          end if;
    end if;;

    call sa_set_http_header('Content-Length', length(res) );
    select res;
end;
```

上記の Web サービスとストアドプロシージャーは、プライマリノード、ミラーノード、コピーノードのロールを持つ任意の SQL Anywhere サーバーに使用できます。ReadOnly データベースプロパティを On に設定すると、**AVAILABLE=TRUE** が返されます。ミラーを変更して、ReadOnly を Off に設定した場合にのみ、writeFarm などの書き込み可能なサーバーファーム内で使用するサービスから **AVAILABLE=TRUE** が返されます。オプションとして、グローバルテーブルのエントリなどの他の要因に基づいて、Outbound Enabler に対する ping サービスの可用性を決めることもできます。たとえば、段階的な配備では、プライマリノードでテーブルやプロシージャーを更新し、ミラーノードとコピーノードに伝達された時点で、各サーバーのサーバーファームメンバーシップをアクティブにしたり非アクティブにしたりします。

### Relay Server 設定

Relay Server を設定するには、次の手順に従います。

1. readFarm と writeFarm の各バックエンドファームを作成します。

2. プライマリノード、ミラーノード、コピーノードを readFarm Relay Server バックエンドファームのメンバーとして設定します。

3. プライマリノードとミラーノードを writeFarm Relay Server バックエンドファームのメンバーとして設定します。

4. どちらのファームも、active_cookie 設定オプションを **yes** に設定し、active_header 設定オプションを **no** に設定します最後の手順を行うには、Sybase Central で **SQL Anywhere** をバックエンドファームの結び付きタイプとして選択します。結び付きを必要とせず、パフォーマンスを向上させる場合、active_cookie 設定オプションを no に設定できます。

### クライアント／アプリケーションの設定

- **制限事項**　現時点では、SQL Anywhere バックエンドサーバーでは、単一のデータベースで稼働している高可用性のみがサポートされています。高可用性のフェールオーバーの最大遅延は、次の項目の最大値になります。

  ○Outbound Enabler のポーリング頻度。「Outbound Enabler の構文」『Relay Server』の -d オプションを参照してください。

  ○ミラーがプライマリになるまでの所要時間。

### 参照

- 「Relay Server 設定ファイル」『Relay Server』
- 「Relay Server ファーム設定の更新」『Relay Server』
- 「バックエンドファームセクションのプロパティ」『Relay Server』

# SQL Anywhere データベースのトラブルシューティング

### 参照

- 「よくある質問 - SQL Anywhere」『SQL Anywhere 12 紹介』
- 「パフォーマンス向上のヒント」『SQL Anywhere サーバー SQL の使用法』
- 「アプリケーションプロファイリング」『SQL Anywhere サーバー SQL の使用法』
- 「SQL Anywhere モニター」1217 ページ
- 「SQL Anywhere コンソールユーティリティ (dbconsole)」905 ページ
- 「パフォーマンスの向上、診断、モニタリング」『SQL Anywhere サーバー SQL の使用法』

## データベースサイズの予期しない変化について

データベースページは、ページのすべてのレコードが削除されると解放されます。解放されたデータベースページは再利用可能になりますが、ファイルから削除できません。ただし、今後 INSERT 文や UPDATE 文で解放されたページを使用できます。

挿入、更新、削除などの変更がデータベースに加えられるたびに、システム DB 領域内に格納されているロールバックログにエントリが追加されます。これらの操作が多数実行されてからコミットが実行されると、ロールバックログは非常に大きくなり、データベースのサイズが増大する可能性があります。

チェックポイントログは、システム DB 領域の最後に格納されています。データベースサーバーが停止すると、チェックポイントログはトランケートされ、システム DB 領域が小さくなります。DELETE または TRUNCATE 操作によって解放されたページは、今後再利用できるようにデータベースファイル内に残り、ファイルから削除できません。

データベースファイルのサイズが増大している場合、または期待どおりに減少しない場合は、次の操作を行います。

- INSERT 文、UPDATE 文、または DELETE 文を使用する場合は、COMMIT を頻繁に実行します。COMMIT が実行されると、ロールバックログに割り付けられたページは、システム DB 領域内で再利用できるように解放されます。

- UPDATE 文または DELETE 文を使用する場合、または INSERT 文を使用し、大きいインデックスが関連する場合は、CHECKPOINT を時々実行します。チェックポイントログ内のページは、各チェックポイント後にチェックポイントログで再利用できるようになります。

- TRUNCATE TABLE を実行します。ページレベルの削除が実行されます。この場合、ページのコピーをチェックポイントログに追加したり、個々のローレベルの操作をロールバックログに追加したりする必要はありません。TRUNCATE TABLE によって解放されたページは、次回のチェックポイントの後でのみ再利用できます。TRUNCATE TABLE を実行すると、次の条件に当てはまる場合、ページレベルの削除が実行されます。

  - トランケートするテーブルへの外部キー、またはテーブルからの外部キーがない。
  - TRUNCATE TABLE がトリガー内で実行されていない。
  - TRUNCATE TABLE がアトミックオペレーションと組み合わせて実行されていない。
  - チェックポイントログページがシステム DB 領域ファイルの最後に書き込まれている。これらのページは、データベースが停止すると削除されます。

データベースの再構築によって、データベースのサイズが減少する場合があります。これは、再構築されたデータベースの空きページが少なくなるためです。

## SQL Anywhere のエラーレポート

SQL Anywhere プログラムで致命的なエラーまたはクラッシュが発生した場合は、エラー発生時のプログラムに関する技術情報を含むエラーレポートが作成されます。エラーレポートが作成されると、このレポートを SQL Anywhere サポートチームに提出するよう要求されます。サポートユーティリティ (dbsupport) を使用して、このレポートを提出することもできます。エラーレポートを送信すると、致命的なエラーやアサーションの原因の究明に役立つことがあります。エラーレポートが提出されると、テクニカルサポートの問い合わせで使用できる提出 ID を受け取ります。問題をテクニカルサポート部門に報告すると、このエラーファイルを送信することで、テクニカルサポート部門が問題をすばやく処理することができます。テクニカルサポートに連絡しない場合でも、製品の改良に役立てるために、エラーレポートを送信することをおすすめします。

レポートにネットワークユーザー名やパスワードなどの機密情報が記載されることのないよう、最大限の注意を払う必要があります。

エラーレポートは、使用するコンピューターのファイルシステムに無期限に格納されます。サポートユーティリティ (dbsupport) を使用して、後日このレポートを送信することもできます。

**♦ エラーレポートを送信するには、次の手順に従います。**

- 次のコマンドを実行します。

```
dbsupport -sa
```

診断ディレクトリに保存されているすべてのクラッシュレポートと診断情報が、iAnywhere Solutions に送信されます。

**参照**

- 「サポートユーティリティ (dbsupport)」1027 ページ

# データベースサーバー起動時のトラブルシューティング

この項では、データベースサーバーの起動時によく発生する問題について説明します。

## トランザクションログファイルが有効であるかを確認する

既存のトランザクションログが無効な場合、データベースサーバーは起動しません。たとえば、開発中に、データベースを新しいバージョンに置き換えると同時にトランザクションログを削除しなかったとします。これは、トランザクションログとデータベースが異なる原因となり、結果として無効なトランザクションログになります。

## テンポラリファイル用の十分なディスク領域があるかを確認する

SQL Anywhere は、テンポラリファイルを使用して実行時に情報を格納します。このファイルは、SATMP 環境変数で指定されるディレクトリ (通常は *c:¥temp*) に保存されます。

テンポラリディレクトリに使用できる十分なディスク領域がないと、サーバーを起動するときに問題が生じることがあります。

**参照**

- 「SATMP 環境変数」451 ページ

## ネットワーク通信の起動時の問題をデバッグする

TCP/IP または通信関係のエラーでサーバーが起動に失敗した場合、-z サーバーデバッグオプションを使用できます。起動情報は、データベースサーバーメッセージウィンドウに表示されます。また、-o オプションを使用して、結果を出力ファイルに記録することもできます。

**参照**

- 「-z dbeng12/dbsrv12 サーバーオプション」296 ページ
- 「-o dbeng12/dbsrv12 サーバーオプション」258 ページ

# データベースサーバーパフォーマンス警告のトラブルシューティング

データベースサーバーのパフォーマンス警告は、データベースサーバーメッセージログに出力されます。これらの警告は、データベースまたはデータベースサーバーでそのパフォーマンスが影響を受ける状態になるとレポートされます。

## パフォーマンス警告：サーバーキャッシュサイズがデータベース "%1" には小さすぎます。

データベースサーバーの最大キャッシュサイズ (データベースサーバーのバッファープールサイズの上限であり、サーバーによって計算されるか、または -ch オプションによって明示的に指定されます) がデータベースの開いている DB 領域すべての合計サイズの 10% 未満である場合に、この警告がレポートされます。

たとえば、21 GB のデータベースを実行しているデータベースサーバーは、サーバーの最大キャッシュサイズが 2.1 GB 未満になると、この警告を発行します。キャッシュサイズはデータベースごとに計算されるため、この警告は、複数のデータベースが同じデータベースサーバーで起動されている場合でも発行されない可能性があります。

**参照**


- 「-ch dbeng12/dbsrv12 サーバーオプション」 208 ページ
- 「パフォーマンス向上のためのキャッシュの使用」『SQL Anywhere サーバー SQL の使用法』
- 「動的キャッシュサイズ決定」『SQL Anywhere サーバー SQL の使用法』
- 「キャッシュサイズをモニタリングする統計」『SQL Anywhere サーバー SQL の使用法』


## パフォーマンス警告：データベース "%3" では、サーバー用に設定された最大値 %2 にページサイズ %1 が一致せず、キャッシュの非効率的な使用の原因となっています。

データベースサーバーのバッファープールのページサイズは、少なくともそのデータベースサーバーで起動されているいずれかのデータベースのページサイズと同じである必要があります。データベースサーバーは、さらに大きいページサイズをキャッシュに使用できます。これは -gp オプションを使用して指定できます。ただし、SQL Anywhere データベースサーバーでは、バッファープール内でデータベースページとページフレームが 1:1 でマッピングされている必要があります。メモリ内の各ページフレームには、1 つのデータベースページのみを格納でき、ページは 1 つのページフレームにのみ表示されます。キャッシュで 8 KB のページが使用され、データベースで 4 KB のページが使用されている場合、メモリ内の各キャッシュページフレームには 4 KB のデータのみが格納され、そのページフレームの残りの 4 KB は無駄になります。この状況では、キャッシュサイズは必要な容量の半分になります。

**参照**


- 「-gp dbeng12/dbsrv12 サーバーオプション」 242 ページ
- 「最大ページサイズの考慮事項」 150 ページ

## パフォーマンス警告：データベース "%2" のテーブル "%1" にユニークなインデックスまたはプライマリキーがありません。

テーブルが更新され、そのテーブルに 11 以上のローが格納されており、プライマリキーまたはユニークなインデックスがない場合に、この警告がレポートされます。

SQL Anywhere では、トランザクションログに論理ローの操作 (UPDATE、INSERT、DELETE) が記録され、物理的なページへの変更は記録されません。トランザクションログ内のロー操作は、プライマリキーまたはユニークなインデックス値によって識別されます。したがって、データベースをリカバリする場合、トランザクションログ内のその操作について更新された値を正しいローに適用できます。プライマリキーまたはユニークなインデックスが存在しない場合は、ローのすべての値がローのプライマリキーとして使用されます。この動作によって、トランザクションログのサイズは大幅に増大します。テーブルにインデックスがない場合は、データベースのリカバリ時間が長くなる可能性があります。

**参照**

- 「トランザクションログサイズの考慮事項」130 ページ
- 「インデックス」『SQL Anywhere サーバー SQL の使用法』
- 「トランザクションログ」127 ページ

## パフォーマンス警告：ページサイズがデータベース "%1" には小さすぎます。

このパフォーマンス警告は、データベースが次の条件をすべて満たしている場合にレポートされます。

1. データベースのページサイズが 2 KB である。
2. データベース全体のページ数が 100000 (データベースサイズで約 200 MB に相当) より多い。

領域の効率とパフォーマンスを向上させるために、データベースをアンロードし、ページサイズが 4 KB 以上の新しいデータベースに再ロードすることをおすすめします。

**参照**

- 「データベースの再構築」『SQL Anywhere サーバー SQL の使用法』
- 「適切なページサイズの使用」『SQL Anywhere サーバー SQL の使用法』

## パフォーマンス警告：データベースファイル "%1" は %2 のディスクフラグメントで構成されています。

データベースサーバーでデータベースが稼働を開始すると、SQL Anywhere は、オペレーティングシステムによってレポートされたメインデータベースファイルのファイルフラグメント数をレポートします。データベースが格納されているディスクボリュームのディスクが過度に断片化されると、I/O 処理中にディスクの遅延時間が長くなる可能性があるため、パフォーマンスの問題が発生することがあります。

ファイルシステムの断片化の問題は、データベース内のテーブルページの断片化とは無関係です。これらの別々の問題を分析して解決するには、それぞれ異なる方法とツールを使用します。ファイルフラグメント数は、Windows ベースの SQL Anywhere データベースサーバーの場合のみレポートされます。UNIX ファイルシステム (Linux ファイルシステムを含む) では、Windows ベースのシステムよりも断片化の問題は大幅に少なくなります。

Windows プラットフォームでは、通常、データベースファイルは複数のフラグメントで構成されています。データベースについてこのメッセージを受け取った場合は、データベースファイルのサイズに対するフラグメント数を確認すると、システム管理者による対処が必要かどうかを判断できます。たとえば、25 個のフラグメントで構成されている 100 GB のデータベースファイルは、深刻な問題になるとは見なされません。ただし、300 個のディスクフラグメントで構成されている 50 MB のデータベースファイルは、パフォーマンスの問題の影響を受ける可能性があります。ファイルのフラグメンテーションの問題を解決するには、データベースをディスクパーティションに単独で入れ、使用可能な Windows ディスクデフラグユーティリティの 1 つを定期的に実行します。

**参照**

- 「断片化の削減」『SQL Anywhere サーバー SQL の使用法』
- 「テーブルの断片化削減」『SQL Anywhere サーバー SQL の使用法』
- 「テーブルの断片化削減」『SQL Anywhere サーバー SQL の使用法』
- DBFileFragments データベースプロパティ 788 ページ

## パフォーマンス警告：データベース "%3" では、テーブル "%1" の AUTOINCREMENT カラム "%2" にインデックスが設定されていません。

この警告は、永久ベーステーブルに AUTOINCREMENT カラムが含まれ、そのカラムにインデックスが設定されていない場合にレポートされます。この警告が生成されるのは、ISYSTABCOL カタログテーブルの MAX_IDENTITY カラムに格納されている、そのカラムの最大 ID 値が不明な場合、または有効でなくなる場合のみです。この状況は、既存のカラムが AUTOINCREMENT カラムに変更された場合、またはデータベースのグローバルオートインクリメント設定が変更された場合に発生する可能性があります。

**参照**

- 「ISYSTABCOL システムテーブル」『SQL Anywhere サーバー SQL リファレンス』
- 「オートインクリメントデフォルト」『SQL Anywhere サーバー SQL の使用法』
- 「GLOBAL AUTOINCREMENT デフォルト」『SQL Anywhere サーバー SQL の使用法』

## パフォーマンス警告：参照オブジェクトの 1 つで DDL 処理が実行されたため、データベース "%2" のビュー "%1" は無効になりました。

この警告は、DDL 処理によって既存のビューが無効になった場合にレポートされます。ビューが変更されたスキーマオブジェクトに依存し、その後ビューの再コンパイルに失敗した場合、そのビューのステータスは INVALID のままです。その後クエリ内からビューが参照されるたび

に、データベースサーバーはビューを再コンパイルしようとします。「依存性とスキーマ変更」『SQL Anywhere サーバー SQL の使用法』を参照してください。

**参照**

- 「ビュー」『SQL Anywhere サーバー SQL の使用法』
- 「通常のビューのステータス」『SQL Anywhere サーバー SQL の使用法』

# ネットワーク通信のトラブルシューティング

ネットワークソフトウェアにはさまざまなコンポーネントが含まれているため、問題が発生しやすくなります。ここで、ネットワークトラブルシューティングのヒントをいくつか説明しますが、ネットワークのトラブルシューティングを支援する一番の情報源は、社内の IT 部門またはネットワーク管理者です。

## ロギングの使用

-z および -o データベースサーバーオプションを指定すると、トラブルシューティング目的で、診断通信メッセージやその他のメッセージがデータベースサーバーメッセージウィンドウに記録されます。クライアントでは、LogFile 接続パラメーターを接続文字列に追加することによって、クライアント側の診断通信メッセージがファイルに記録されます。これらのメッセージを利用して、接続に失敗したときの状況、接続試行に使用された接続パラメーター、使用された通信リンクを特定できます。

**参照**

- 「-z dbeng12/dbsrv12 サーバーオプション」296 ページ
- 「LogFile (LOG) 接続パラメーター」354 ページ

## タイムアウト値の調整

接続が予期せずに切断してしまう場合は、活性タイムアウトまたはアイドルタイムアウトの値を調整することを検討してください。

**参照**

- 「LivenessTimeout (LTO) 接続パラメーター」352 ページ
- 「-tl dbeng12/dbsrv12 サーバーオプション」278 ページ
- 「Idle 接続パラメーター」348 ページ
- 「-ti dbeng12/dbsrv12 サーバーオプション」277 ページ

# データベースのモニタリング

この項では、SQL Anywhere モニターを使用して SQL Anywhere データベースと Mobile Link サーバーをモニターする方法について説明します。また、SQL Anywhere SNMP Extension Agent の設定方法についても説明します。

## SQL Anywhere モニター

SQL Anywhere モニターは、SQL Anywhere データベース (読み込み専用スケールアウトシステムとミラーリングシステム内のデータベースを含む)、Mobile Link サーバー、Mobile Link サーバーファーム、Relay Server ファームの正常性や可用性に関する情報を示すブラウザーベースの管理ツールであり、単に「モニター」とも呼ばれます。

この項では、モニターを使用して **SQL Anywhere** データベース、**MobiLink サーバー**、**Relay Server ファーム**に関するメトリックを収集する方法について説明します。

モニターには次の機能があります。

- **データの常時収集**　SQL Anywhere に対応している他の多くの管理ツールとは異なり、SQL Anywhere モニターはユーザーがブラウザーでログインしていない時も含めてメトリックを常時収集します。また、シャットダウンされるまでメトリックの収集を続けます。
- **電子メールによる警告の通知**　メトリックが収集されると、モニターでは、収集したメトリックを検証し、リソースの異常を示す状況を検出した場合に、警告の電子メールを送信します。
- **ブラウザーベースのインターフェイス**　いつでもブラウザーを使用してモニターにログインし、収集された警告やメトリックを表示できます。
- **複数のデータベース、Mobile Link サーバー、Mobile Link サーバーファーム、Relay Server ファームのモニタリング**　1 つのツールで、同一または異なるコンピューターで実行されている複数のリソースを同時にモニタリングできます。
- **パフォーマンスへの影響を最小化**　モニタリングによってパフォーマンスが低下することはないため、開発環境や運用環境でモニターを日常的に使用できます。

モニターは、次のタスクを担当するユーザーを、そのユーザーが DBA かどうかに関係なく支援するよう設計されています。

- データベース、Mobile Link サーバー、または Relay Server ファームがネットワークに接続されていることを確認する。
- データベース、Mobile Link サーバー、または Relay Server ファームに十分なディスク領域またはメモリがあることを確認する。
- ユーザーがブロックされていないこと、またはクエリの実行に長時間かかっていないことを確認する。
- 所定の期間内に Mobile Link サーバーが実行した同期の回数を確認する。

●Relay Server がトラフィックを送る準備ができたことを確認する。

●Relay Server ファームのトラフィック量、スループット、失敗率を確認する。

**参照**

SQL Anywhere データベースに対して使用できる他の管理ツールやパフォーマンスツールの詳細については、次の項を参照してください。

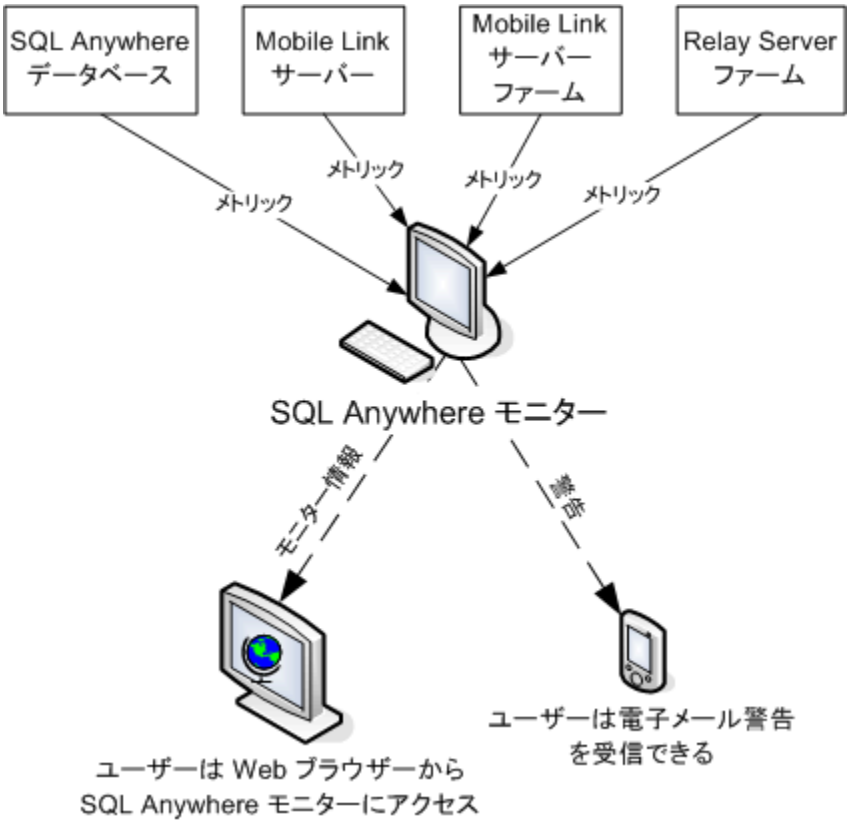

- 「アプリケーションプロファイリング」『SQL Anywhere サーバー SQL の使用法』
- 「SQL Anywhere コンソールユーティリティ (dbconsole)」1021 ページ
- 「統計をモニタリングするための Sybase Central パフォーマンスモニター機能」『SQL Anywhere サーバー SQL の使用法』
- 「Windows パフォーマンスモニターを使用した統計値のモニタリング」『SQL Anywhere サーバー SQL の使用法』
- 「パフォーマンス統計値ユーティリティ (dbstats) (UNIX)」991 ページ


Mobile Link サーバーに対して使用できる他の管理ツールやパフォーマンスツールの詳細については、「Mobile Link モニター」『Mobile Link サーバー管理』を参照してください。

## モニターのアーキテクチャー

モニターは、他のコンピューター上で実行されている SQL Anywhere データベース (読み込み専用スケールアウトシステムとミラーリングシステムのデータベースを含む)、Mobile Link サーバー、Mobile Link サーバーファーム、Relay Server ファームからメトリックやパフォーマンスデータを収集します。その間、別のコンピューターがブラウザー経由でモニターにアクセスします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1




モニターは、SQL Anywhere の組み込みの HTTP サーバーを介して Web ブラウザーで実行される Flash ベースのアプリケーションです。モニターとの対話は、ブラウザーインターフェイス経由で行います。

モニターには、次の 2 つのエディションがあります。

- **SQL Anywhere モニター Developer Edition**　モニター Developer Edition は、開発とテストの目的で使用されます。モニター Developer Edition はデフォルトで SQL Anywhere とともにインストールされ、インストールされた SQL Anywhere をバックエンドで使用します。

- **SQL Anywhere モニター Production Edition**　Production Edition は、配備と運用の目的で使用されます。別個にインストールされ、サービスとして実行されます。Production Edition には、すべての機能が搭載された SQL Anywhere インストールが含まれます。このエディションは、SQL Anywhere の特定のエディションでのみ使用できます。「運用環境での SQL Anywhere モニターの実行」1312 ページを参照してください。

## 前提条件

### モニターがインストールされているコンピューターの場合

○Windows ファイアウォールが有効になっている Windows システムで SQL Anywhere モニターを実行するには、ポート 4950 のポート例外を追加する必要があります。

○SQL Anywhere モニターは、起動時に 384 MB の仮想アドレス領域を予約します。Linux でモニターを起動する場合は、少なくともこの容量の仮想領域が使用可能であることを確認してください。

○ECC 暗号化または FIPS 認定の暗号化を使用して保護されたリソースをモニタリングするには、別途ライセンスが必要です。「モニター通信のセキュリティ」1316 ページを参照してください。

○モニターはモニタリング対象のリソースと同じコンピューター上で実行できますが、特に運用環境では、データベースサーバー、Mobile Link サーバー、Relay Server ファーム、またはその他のアプリケーションへの影響を最小限に抑えるために、モニターを別のコンピューター上で実行することをおすすめします。

○運用環境では、モニター Production Edition を使用することをおすすめします。「運用環境での SQL Anywhere モニターの実行」1312 ページを参照してください。

○モニター Developer Edition を実行している場合は、SQL Anywhere 12.0.0 以降がインストールされている必要があります。モニター Developer Edition では、インストールされた SQL Anywhere をバックエンドで使用します。

**モニターにアクセスするコンピューターの場合**

○オペレーティングシステムに対応している Adobe Flash Player の最新バージョンをインストールします。モニターには、Adobe Flash Player のバージョン 10 との下位互換性があります。適切なバージョンを確認するには、次の URL を参照してください。

○ブラウザーで JavaScript を有効にします。

○SQL Anywhere モニターへのアクセスに使用するブラウザーが実行されているコンピューターが、モニターがインストールされているネットワークに接続されていることを確認します。

モニターには、モニターが実行されているのと同じコンピューターからアクセスできますが、特に運用環境では、別のコンピューターからアクセスすることをおすすめします。

## 制限事項

●モニターを使用してメトリックを収集できる SQL Anywhere データベース、Mobile Link サーバー、Mobile Link サーバーファームのバージョンは次のとおりです。

○SQL Anywhere 10.0.0、10.0.1、11.0.0、11.0.1、12.0.0、12.0.1

○最初の EBF 以降が適用された Mobile Link 11.0.0、11.0.1、12.0.0、12.0.1

○Relay Server 12.0.0、12.0.1

●コンピューター上で一度に実行できるモニターのエディションは 1 つだけです。

●Linux では、インストールしたユーザーだけがモニター Developer Edition を実行できます。

●Linux では、管理者権限を持つユーザーだけがモニター Production Edition をインストールし、実行できます。

- モニターは、TCP/IP 経由で SQL Anywhere データベースに接続します。ローカルデータベースサーバー (dbeng12) 上で実行中のデータベースをモニタリングするには、データベースサーバーを -x all または -x tcpip オプションで起動します。

- クエリの最適化やアプリケーションの速度の測定には使用できません。データベースやアプリケーションのパフォーマンスをチューニングするには、**アプリケーションプロファイリングウィザード**、**Sybase Central パフォーマンスモニター**、または **Windows パフォーマンスモニター**などのツールを使用します。

- 個々の同期に関する情報は提供しません。タイミングなどの同期別の統計を含む、個々の同期に関する詳細情報を取得するには、Mobile Link モニターを使用します。「Mobile Link モニター」『Mobile Link サーバー管理』を参照してください。

**参照**

- 「モニターを使用するためのクイックスタート」1221 ページ

# モニターを使用するためのクイックスタート

SQL Anywhere データベース、Mobile Link サーバー、Mobile Link サーバー、および Relay Server ファームのモニタリングの設定手順は次のとおりです。

1. モニタリングするデータベース、Mobile Link サーバー、または Relay Server ファームを起動します (実行されていない場合)。

2. ネットワークに常時接続されるコンピューターにモニターをインストールします。

   SQL Anywhere モニターはモニターリング対象のリソースと同じコンピューター上で実行できますが、特に運用環境では、モニターを別のコンピューター上で実行して、リソースまたはその他のアプリケーションへの影響を最小限に抑えることをおすすめします。

3. SQL Anywhere モニターを起動し、ブラウザーで開きます。「モニターの起動」1235 ページを参照してください。

4. モニターに管理者としてログインします。デフォルトのユーザーは、名前 **admin** とパスワード **admin** を持つ管理者ユーザーです。

   **注意**
   次のタスクを実行するには、モニターに管理者としてログインしてください。「モニターのユーザー」1296 ページを参照してください。

5. 管理者として左のナビゲーションメニューで、**[ツール]** » **[管理]** をクリックして、モニタリング対象のリソースを追加します。「リソースの追加」1249 ページを参照してください。

6. 管理者は、新規ユーザーを追加し、admin ユーザーのパスワードを変更します。「モニターユーザーの作成」1298 ページを参照してください。

7. 管理者は、リソースに対する警告のスレッショルドを設定します。「警告」1301 ページを参照してください。

8. 管理者は、バックアップとメンテナンスのスケジュールを設定します。「モニターのバックアップ」1309 ページを参照してください。

9. **[閉じる]** をクリックして **[管理]** ウィンドウを終了します。

10. **[概要]** をクリックします。**[リソースリスト]** ウィジェットに、モニタリング対象のリソースが表示されます。「[概要] ダッシュボード」1240 ページを参照してください。

# チュートリアル：モニターによるリソースのモニタリング

このチュートリアルを参考にして、SQL Anywhere データベース、Mobile Link サーバーおよび Mobile Link サーバーファームまたは Relay Server ファームのモニタリングを設定してください。このチュートリアルでは、モニター Developer Edition を使用しています。

## レッスン 1：デフォルトの管理者としてモニターにログインする

**♦ モニターを起動してログインするには**

1. モニターを起動します。次の手順では、モニターが現在バックグラウンドで実行されていないことを前提としています。

   **モニター Developer Edition を起動するには、次の手順に従います (Windows の場合)。** **[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[SQL Anywhere モニター]** をクリックします。

   **モニター Developer Edition を起動するには、次の手順に従います (Linux の場合)。** モニターのインストールディレクトリの *bin32* または *bin64* ディレクトリから *samonitor.sh* スクリプトを実行します。

   ```
   ./samonitor.sh launch
   ```

   モニターによってメトリックの収集が開始され、モニターにログインするためのデフォルト URL *http://localhost:4950* がブラウザーに表示されます。

   > **注意**
   > ネットワークを経由してモニターにアクセスしている場合は、**http://***computer-name***:4950** をブラウズします。*computer-name* は、モニターが実行されているコンピューターの名前です。

2. デフォルトの管理者ユーザーとしてモニターにログインします。

   **[ユーザー名]** フィールドに **admin** と入力し、**[パスワード]** フィールドに **admin** と入力します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**注意**
チュートリアルで次のレッスンを実行するには、モニターに管理者としてログインしてください。読み込み専用ユーザーとオペレーターユーザーにはすべてのタスクを実行する権限がありません。

**♦ モニターユーザータイプの確認**

1. モニターにログインします。
2. **[ツール]** » **[ユーザー設定]** をクリックし、**[ユーザーのタイプ]** 設定を確認します。

「モニターのユーザー」1296 ページを参照してください。

デフォルトでは、次の 2 つのウィジェットがある **[概要]** ダッシュボードが開きます。

1. **[リソースリスト]** ウィジェットには、モニタリング対象のリソースがリストされます。モニターを初めて開いたときは、**[SQL Anywhere モニター]** という名前のデフォルトのリソースを使用してモニター自体のみがモニタリングされています。このリソースを変更したり、このリソースのモニタリングを停止することはできません。
2. **[警告リスト]** ウィジェットには、モニタリング対象のリソースからの警告がリストされます。

**参照**

- 「モニターの起動」1235 ページ

# レッスン 2：リソースをモニタリングするためのモニターの設定

このレッスンでは、リソースをモニタリングするためのモニターを設定します。次のうちの 1 つを選択してください。

- 「SQL Anywhere のモニタリング」
- 「Relay Server ファームのモニタリング」
- 「レッスン 13：SQL Anywhere モニターでの Mobile Link リソースのモニター」『Mobile Link クイックスタート』

## SQL Anywhere のモニタリング

SQL Anywhere サンプルデータベース *demo.db* を起動し、このデータベースをモニタリング対象のリソースとして追加します。

**♦ データベースリソースをモニターに追加するには、次の手順に従います。**

1. SQL Anywhere サンプルデータベースを起動します。**[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[SQL Anywhere]** » **[ネットワークサーバーのサンプル]** をクリックします。
2. モニターに管理者としてログインします。「レッスン 1：デフォルトの管理者としてモニターにログインする」1222 ページを参照してください。
3. 左のサイドバーにある **[ツール]** » **[管理]** をクリックします。
4. **[リソース]** をクリックして、**[追加]** をクリックします。
5. **[SQL Anywhere サーバー]** をクリックして **[次へ]** をクリックします。
6. **[名前]** フィールドに **demo12** と入力し、**[次へ]** をクリックします。
7. **[ホスト]** フィールドに **localhost** と入力し、**[サーバー]** フィールドに **demo12** と入力します。
8. **[作成]** をクリックします。
9. 必要な認証情報が要求されたら、**[DBA ユーザー ID]** フィールドに **DBA** と入力し、**[パスワード]** フィールドに **sql** と入力します。**[OK]** をクリックします。

   データベースをモニタリング対象のリソースとして追加する場合は、データベースの DBA ユーザー ID とパスワードを指定する必要があります。これらのクレデンシャルは次の操作を行う場合に使用されます。

   - データベースに接続します。
   - **sa_monitor_user** という名前の新しいユーザーを作成します。モニターは、sa_monitor_user を使用してデータベースに接続し、モニタリングします。
   - データベースをモニタリングするために **sa_monitor_user** が必要とするデータベースオブジェクトをインストールします。インストールされるオブジェクトのリストについては、「モニターがインストールしたデータベースオブジェクトのリスト」1311 ページを参照してください。
   - モニターから DBA クレデンシャルを破棄します。**sa_monitor_user** はモニタリングされるデータベースに追加されるので、そのデータベース外部のどこかに DBA クレデンシャル

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

を格納する必要はありません。「モニターがインストールしたデータベースオブジェクトのリスト」1311 ページを参照してください。

リソースが追加され、モニタリングが自動的に開始されます。

10. **[閉じる]** をクリックします。

11. **[閉じる]** をクリックします。

12. **[ダッシュボード]** » **[概要]** をクリックします。

    **[リソースリスト]** に **demo12** リソースが表示されます。

13. **demo12** をクリックすると、demo12 ダッシュボードが開き、収集されたメトリックが表示されます。

14. 「レッスン 3：警告のテスト」1226 ページに進みます。

### MobiLink サーバーまたは MobiLink サーバーファームのモニタリング

MobiLink サーバーまたは MobiLink サーバーファームリソースのモニタリングについては、「レッスン 13：SQL Anywhere モニターでの Mobile Link リソースのモニター」『Mobile Link クイックスタート』 を参照してください。

### Relay Server ファームのモニタリング

このレッスンでは、Relay Server ファームを起動し、モニタリング対象のリソースとして追加します。

**♦ リソースをモニターに追加するには、次の手順に従います。**

1. Relay Server ファームを起動します。

2. モニターに管理者としてログインします。「レッスン 1：デフォルトの管理者としてモニターにログインする」1222 ページを参照してください。

3. **[ツール]** » **[管理]** をクリックします。

4. **[リソース]** をクリックして、**[追加]** をクリックします。

5. **[Relay Server ファーム]** を選択して **[次へ]** をクリックします。

6. **[名前]** フィールドに、Relay Server ファームリソースの名前を入力し、**[次へ]** をクリックします。

7. **[ホスト]** フィールドに、Relay Server が実行されているコンピューターのホスト名または IP アドレスを入力し、**[次へ]** をクリックします。

   この項目は必須です。「Host 接続パラメーター」346 ページを参照してください。

8. **[作成]** をクリックします。

   新しいリソースが作成され、モニタリングが開始されます。

9. **[閉じる]** をクリックします。

10. **[閉じる]** をクリックします。

11. **[概要]** » **[リソースリスト]** をクリックします。リソースをクリックしてリソースのダッシュボードを作成し、開きます。

12. 「レッスン 3：警告のテスト」1226 ページに進みます。

## レッスン 3：警告のテスト

このレッスンでは、警告の処理を予行練習するために、警告を意図的にトリガーします。

♦ **警告を表示および解決するには、次の手順に従います。**

1. 警告をトリガーします。トリガーするには、データベースサーバー、Mobile Link サーバー、または Relay Server をシャットダウンします。

   データベースサーバーの例：

   a. Windows では、システムトレイで **demo12** データベースサーバーのネットワークサーバーアイコンをダブルクリックします。

   b. データベースサーバーメッセージウィンドウで **[シャットダウン]** をクリックします。

   c. **[はい]** をクリックします。

   Mobile Link サーバー：

   a. Windows では、システムトレイで **MobiLinkServerSample** Mobile Link サーバーの Mobile Link サーバーアイコンをダブルクリックします。

   b. Mobile Link サーバーメッセージウィンドウで **[シャットダウン]** をクリックします。

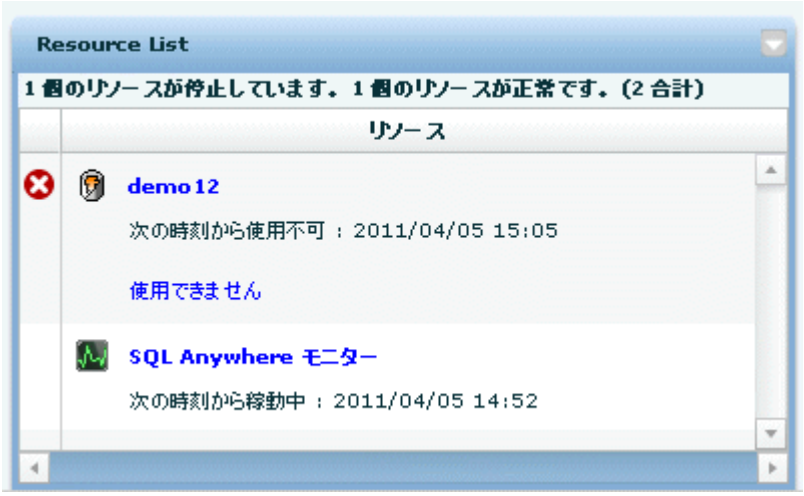
   c. **[はい]** をクリックします。

2. モニターに管理者としてログインします。「レッスン 1：デフォルトの管理者としてモニターにログインする」1222 ページを参照してください。

3. **[ダッシュボード]** » **[概要]** をクリックします。

   **[リソースリスト]** では、**demo12** リソース、**MobiLinkServerSample** リソース、または Relay Server ファームリソースの隣に、中に白い X がある赤い円が表示されます。このアイコンは、リソースが使用不可であることを示します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1




**[警告リスト]** ウィジェットでは、**[使用できません]**、**[サーバーが停止しています]** 警告、または**[Relay Server の停止が持続しています]** 警告が表示され、そのステータスは **[アクティブ]** です。

状態やモニターの状態が変わるまで数秒かかることがあります。デフォルトでは、リソースの情報は 30 秒ごとに収集されます。「収集間隔」1263 ページを参照してください。

4. **[警告リスト]** ウィジェットでは、データベースの **[使用できません]**、MobiLink の **[サーバーが停止しています]**、または Relay Server の **[Relay Server の停止が持続しています]** をクリックしてから、**[詳細]** をクリックして説明を表示します。

5. **[OK]** をクリックして、警告を閉じます。

6. サンプルデータベース、同期サーバーのサンプル、または Relay Server ファームを再起動します。

♦ **サンプルデータベースを再起動します。**

● **[スタート] » [プログラム] » [SQL Anywhere]12 » [SQL Anywhere] » [ネットワークサーバーのサンプル]** をクリックします。

♦ **同期サーバーのサンプルに移動します。**

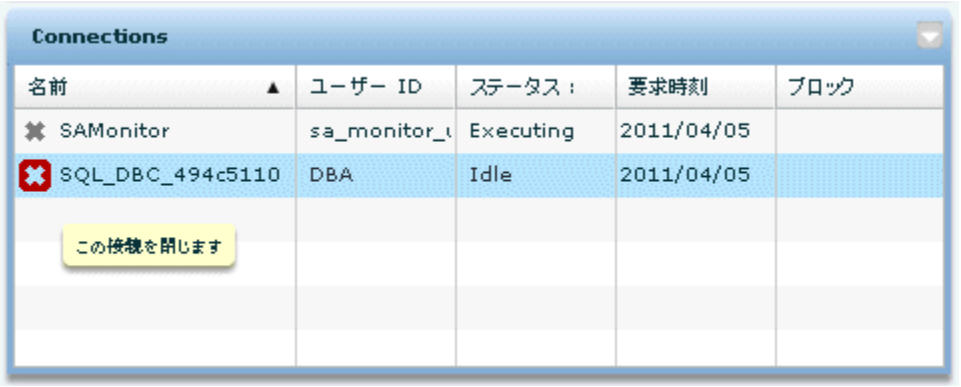
● **[スタート] » [プログラム] » [SQL Anywhere 12] » [Mobile Link] » [同期サーバーのサンプル]** をクリックします。

モニターでは、**[リソースリスト]** で、リソースの **[ステータス]** は黄色の三角に変わります。このアイコンは、リソースがモニタリングされており、警告が出されていることを示します。**[警告リスト]** の**[使用できません]**、**[サーバーが停止しています]** 警告、または **[Relay Server の停止が持続しています]** 警告の **[ステータス]** は **[非アクティブ]** に変わります。非アクティブの警告は、その警告をトリガーした問題は存在しませんが、警告が解決または削除されていないことを示しています。

7. 警告を解決します。解決するには、目的の警告を選択し、**[解決済みとマークする]** をクリックします。

> **注意**
> 警告を解決して削除できるのは、管理者とオペレーターだけです。「モニターのユーザー」1296 ページを参照してください。

現在、**[リソースリスト]** には、**demo12** リソース、**MobiLinkServerSample** リソース、または Relay Server ファームリソースの隣にアイコンがないので、リソースはモニタリングされており、警告がないことがわかります。**[警告リスト]** では、**[使用できません]**、**[サーバーが停止しています]** 警告、、または **[Relay Server の停止が持続しています]** 警告の **[ステータス]** が **[*admin* によって解決済み]** に変わっています。ここで、*admin* はモニターのユーザー名を指します。




8. 警告を削除します。
   a. 警告をクリックして、**[削除]** をクリックします。
   b. **[選択した警告を削除する]** をクリックしてから、**[OK]** をクリックします。
   c. プロンプトが表示されたら、**[はい]** をクリックして、削除を確認します。

9. 「レッスン 4：警告発生時に電子メールが送信されるようにするための設定」1229 ページに進みます。

## レッスン 4：警告発生時に電子メールが送信されるようにするための設定

発生した警告は、特定のリソースのダッシュボードの **[警告リスト]** ウィジェットに必ず表示されます。「レッスン 3：警告のテスト」1226 ページを参照してください。

このレッスンでは、警告が発生したら必ず電子メールが送信されるようにモニターを設定します。

**♦ 電子メールによる通知を設定するには、次の手順に従います。**

1. モニターに管理者としてログインします。「レッスン 1：デフォルトの管理者としてモニターにログインする」1222 ページを参照してください。

2. 電子メールを受信するユーザーを追加します。
   a. **[ツール]** » **[管理]** をクリックします。
      プロンプトが表示されたら、パスワードとして **admin** を入力します。
   b. **[ユーザー]** をクリックしてから、**[新規]** をクリックします。
   c. **[ユーザー名]** フィールドにユーザー名を入力します。
   d. **[パスワード]** フィールドと **[パスワードの確認]** フィールドに、パスワードを入力します。
   e. **[電子メール]** フィールドに、有効な電子メールアドレスを入力します。
   f. **[ユーザーのタイプ]** として、**[オペレーター]** をクリックします。

オペレーターは、電子メールで警告を受信したり、警告の解決と削除を実行できます。このユーザーは、ほとんどのモニターのウィジェットにアクセスできますが、**[管理]** ウィンドウにはアクセスできません。

g. **[次へ]** をクリックします。

h. 対象のリソースを選択するよう要求されたら、**[すべてオン]** をクリックします。

i. **[保存]** をクリックします。

新しいユーザーが作成され、**[管理]** ウィンドウに戻ります。

j. **[閉じる]** をクリックします。

3. 電子メールによる警告の通知を設定します。

> **注意**
> 次のタスクを実行するには、モニターに管理者としてログインしてください。モニターで電子メールの送信を設定できるのは管理者のみです。「モニターのユーザー」1296 ページを参照してください。

a. **[管理]** ウィンドウで、**[設定]** をクリックし、**[編集]** をクリックします。

b. **[警告の通知]** タブで、**[警告の通知を電子メールで送信]** をクリックします。

c. その他の設定を必要に応じて指定します。「モニターによる警告電子メールの送信の有効化」1307 ページを参照してください。

d. 電子メールによる通知が適切に設定されたかどうかをテストします。

**[テスト電子メールを送信]** をクリックします。

e. 画面の要求に従い、テスト電子メールの送信先電子メールアドレスを入力して、**[OK]** をクリックします。

テスト電子メールが、指定された電子メールアドレスに送信されます。

f. **[保存]** をクリックします。

g. **[閉じる]** をクリックします。

4. 「レッスン 5：新しいダッシュボードとウィジェットの追加」1230 ページに進みます。

警告が発生すると、指定されたユーザー宛に、その警告に関する情報が記載された電子メールが送信されます。

## レッスン 5：新しいダッシュボードとウィジェットの追加

ダッシュボードは、各ユーザーに固有です。どのユーザーも各自のダッシュボードを追加、編集、または削除できます。また、どのユーザーも各自のダッシュボードにあるウィジェットを追加、編集、または削除できます。デフォルトでは、ダッシュボードの作成時にデフォルトのウィジェットが取り込まれます。このレッスンでは、モニターにダッシュボードを追加し、次にウィジェットを追加します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**♦ 新しいダッシュボードを追加するには、次の手順に従います。**

1. モニターにログインします。
2. **[ダッシュボード]** » **[新規追加]** をクリックします。
3. **[次のリソースをモニターするダッシュボード]** をクリックし、リソースを選択します。
4. **[ダッシュボード名]** フィールドに、「user_joe」 と入力します。
5. **[カラムの数]** フィールドに、「2」 と入力します。
6. **[OK]** をクリックします。

   新しいダッシュボードが、次のウィジェットとともに表示されます。

   **SQL Anywhere** **[警告リスト]**、**[主要なパフォーマンスのメトリック]**、**[サーバー情報]**、**[接続]** が表示されます。

   **MobiLink** **[警告リスト]**、**[主要なパフォーマンスのメトリック]**、**[サーバー情報]**、**[接続]** が表示されます。

   **Relay Server** **[警告リスト]** と **[主要なパフォーマンスのメトリック]** が表示されます。

ここでは、デフォルトのスパークラインではなく、グラフを表示する新しいメトリックウィジェットを作成する方法について説明します。

**♦ グラフがあるメトリック表示ウィジェットを追加するには、次の手順に従います。**

1. **user_joe** のダッシュボードの右上で、**[カスタマイズ]** をクリックし、**[ウィジェットの追加]** をクリックします。
2. **[メトリック]** をクリックして、**[次へ]** をクリックします。
3. **[このウィジェットの名前を指定してください。]** フィールドに **Metrics display** と入力します。
4. **[どのようなリソースに関心がありますか ?]** フィールドで、データベースサーバーの場合は **demo12**、MobiLink サーバーまたは Relay Server の Relay Server ファームリソースの場合は **MobiLinkServerSample** をクリックします。
5. **[どのような表示が必要ですか ?]** フィールドで、**[グラフ]** をクリックします。
6. **[どのようなメトリックを確認したいですか ?]** で、次をクリックします。

   **SQL Anywhere** **[CPU 使用率]**、**[メモリのメトリック]** » **[キャッシュサイズ]**、および **[処理されたクエリ]**

   **MobiLink** 同期のメトリック

   **Relay Server** **[CPU 使用率]**

7. **[作成]** をクリックします。

   ダッシュボードにウィジェット **Metrics display** が表示されます。

   ウィジェットのサイズを最大にするには、ウィジェットウィンドウ枠でドロップダウンメニューの矢印をクリックし、**[最大化]** をクリックします。

   グラフの詳細を確認するには、カーソルをグラフの特定の位置に置きます。

8. 「レッスン 6：データベース接続を閉じる」1232 ページに進みます。

**参照**

- 「ダッシュボード」1243 ページ
- 「ウィジェット」1244 ページ

## レッスン 6：データベース接続を閉じる

このレッスンでは、モニターを使用してリソースデータベースへの接続を閉じます。接続を閉じるには、リソースデータベースの DBA ユーザー ID とパスワードが必要です。

**♦ データベース接続を閉じるには、次の手順に従います。**

1. モニターに管理者としてログインします。「レッスン 1：デフォルトの管理者としてモニターにログインする」1222 ページを参照してください。

2. **demo12** ダッシュボードを開きます。

3. **接続** ウィジェットで、閉じる接続の横にある **x** をクリックします。

4. 必要な認証情報が要求されたら、**[DBA ユーザー ID]** フィールドに **DBA** と入力し、**[パスワード]** フィールドに **sql** と入力します。

   モニターでは、DBA ユーザー ID とパスワードを使用して、接続を削除するリソースデータベースに接続します。DBA ユーザー ID とパスワードはモニターによって保持されません。

5. **[OK]** をクリックします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**Connections** ウィジェットから接続が削除されます。

6. 「レッスン 7 : (オプション) 読み込み専用スケールアウトシステムの一部であるデータベースをモニタリングするためのモニターの設定」1233 ページに進みます。

## レッスン 7 : (オプション) 読み込み専用スケールアウトシステムの一部であるデータベースをモニタリングするためのモニターの設定

読み込み専用スケールアウトシステムでルートノートである SQL Anywhere データベースをモニタリングするには、「チュートリアル : 読み込み専用のスケールアウトシステムの作成とモニタリング」1190 ページを参照してください。

「レッスン 8 : (任意) ミラーリングシステムの一部であるデータベースをモニタリングするためのモニターの設定」1233 ページに進みます。

## レッスン 8 : (任意) ミラーリングシステムの一部であるデータベースをモニタリングするためのモニターの設定

ミラーリングシステムのプライマリサーバーである SQL Anywhere データベースをモニタリングするには、「チュートリアル : データベースミラーリングの使用」1146 ページを参照してください。

「レッスン 9 : クリーンアップ」1233 ページに進みます。

## レッスン 9 : クリーンアップ

このレッスンでは、モニタリングしているリソースを削除します。これにより、収集されたメトリックが削除され、データの収集が停止します。リソースのモニタリングを継続する必要がある運用環境では、データベース、MobiLink サーバー、または Relay Server ファームとモニターを実行させたままにします。

このレッスンでは、Relay Server リソースを削除します。これにより、収集されたメトリックが削除され、データの収集が停止します。Relay Server ファームのモニタリングを継続する必要がある運用環境では、Relay Server ファームとモニターをどちらも実行させたままにします。

> **注意**
> 次のタスクを実行するには、モニターに管理者としてログインしてください。リソースを削除できるのは管理者だけです。**SQL Anywhere モニター**リソースは削除できません。「モニターのユーザー」1296 ページを参照してください。

**♦ リソースのモニタリングを停止します。**

1. モニターに管理者としてログインします。

2. **demo12** (SQL Anywhere の場合)、**MobiLinkServerSample** (MobiLink サーバーの場合)、または Relay Server ファームリソース (Relay Server の場合) を削除します。

a. **[ツール]** » **[管理]** をクリックします。

b. **[リソース]** をクリックします。

c. **demo12**、**MobiLinkServerSample**、または Relay Server ファームリソースをクリックし、**[停止]** をクリックします。

d. **[削除]** をクリックします。

e. **[はい]** をクリックして、リソースの削除を確定します。

> **注意**
> データベースリソースを削除しても、データベースにインストールされたモニタリングオブジェクトは**削除されません。**これらのオブジェクトを削除するには、DBA 権限を持つユーザーとしてデータベースに接続し、次のコマンドを実行する必要があります。
>
> ```
> DROP USER sa_monitor_user;
> ```
>
> 「モニタリングオブジェクトの削除」1312 ページを参照してください。

f. **[閉じる]** をクリックします。

3. モニターからログアウトします。

   **[ログアウト]** をクリックします。

4. SQL Anywhere モニターを表示していたブラウザーウィンドウを閉じます。

5. モニターを終了し、モニタリングを停止します。

   ●Windows では、システムトレイで [SQL Anywhere モニター] アイコンを右クリックし、**[SQL Anywhere モニターの終了]** をクリックします。

   ●Linux では、モニターのインストールディレクトリの *bin32* または *bin64* ディレクトリから *samonitor.sh* スクリプトを実行します。

   ```
   ./samonitor.sh stop
   ```

   モニターは、すべてのリソースについてメトリックの収集を停止します。

6. 次のいずれかを行います。

   ●SQL Anywhere データベースをシャットダウンします。

   a. システムトレイで demo12 データベースサーバーのネットワークサーバーアイコンをダブルクリックします。

   b. データベースサーバーメッセージウィンドウで **[シャットダウン]** をクリックします。

   c. **[はい]** をクリックします。

   ●Mobile Link サーバーを停止します。

   a. システムトレイで Mobile Link 同期サーバーのサンプルの [Mobile Link サーバー] アイコンをダブルクリックします。

   b. Mobile Link サーバーメッセージウィンドウで **[シャットダウン]** をクリックします。

●Relay Server ファームをシャットダウンします。

# モニターの起動

モニターを起動すると、モニターではリソースに接続し、指定されているすべてのリソースのメトリックを収集します。

モニターは何も通知せずにバックグラウンドで稼働するように設計されています。モニターとの対話は、ブラウザーインターフェイス経由で行います。メトリックを収集するには、バックグラウンドでモニターの実行を継続することをおすすめします。

**♦ モニター Developer Edition の起動 (Windows の場合)**

> **注意**
> コンピューター上で一度に実行できるモニターのエディションは 1 つだけです。

1. モニターがインストールされているコンピューターで、**[スタート]** » **[プログラム]** » **[SQL Anywhere 12]** » **[管理ツール]** » **[SQL Anywhere モニター]** をクリックします。

   モニターはリソースに接続し、モニターに指定されているすべてのリソースのメトリックの収集を開始します。

   ●ブラウザーに、モニターにログインするためのデフォルト URL *http://localhost:4950* が表示されます。

   > **注意**
   > ネットワークを経由してモニターにアクセスしている場合は、**http:**//*computer-name*:**4950** をブラウズします。*computer-name* は、モニターが実行されているコンピューターの名前です。「モニターにリモートでログイン」1239 ページを参照してください。

   ●[SQL Anywhere モニター] アイコンがシステムトレイに表示されます。

2. ログインします。

   ブラウザーで、モニターのユーザー名とパスワードを入力します。モニターのユーザー ID とパスワードでは、大文字と小文字が区別されます。ユーザーには、管理者、オペレーター、および読み込み専用ユーザーの 3 つの異なるタイプがあります。それぞれのユーザータイプには、異なるパーミッションが付与されています。デフォルトのユーザーは、名前 **admin** とパスワード **admin** を持つ管理者です。「モニターのユーザー」1296 ページを参照してください。

   **[このコンピューターにログイン情報を記憶]** をクリックし、2 週間またはログアウトするまでセッションが維持されるようにできます。**[このコンピューターにログイン情報を記憶]** をオフにすると、ブラウザーを閉じるか、ログアウトしたときにセッションの有効期限が切れます。

### ♦ モニター Developer Edition の起動 (Linux の場合)

> **注意**
> コンピューター上で一度に実行できるモニターのエディションは 1 つだけです。

1. モニターがインストールされているコンピューターで、モニターを起動します。

   モニターのインストールディレクトリの *bin32* または *bin64* ディレクトリから *samonitor.sh* スクリプトを実行します。

   ```
   ./samonitor.sh launch
   ```

   モニターはリソースに接続し、モニターに指定されているすべてのリソースのメトリックの収集を開始します。ブラウザーに、モニターにログインするためのデフォルト URL *http://localhost:4950* が表示されます。

   > **注意**
   > ネットワークを経由してモニターにアクセスしている場合は、**http:**//*computer-name***:4950** をブラウズします。*computer-name* は、モニターが実行されているコンピューターの名前です。「モニターにリモートでログイン」1239 ページを参照してください。

2. ログインします。

   ブラウザーで、モニターのユーザー名とパスワードを入力します。モニターのユーザー ID とパスワードでは、大文字と小文字が区別されます。ユーザーには、管理者、オペレーター、および読み込み専用ユーザーの 3 つの異なるタイプがあります。それぞれのユーザータイプには、異なるパーミッションが付与されています。デフォルトのユーザーは、名前 **admin** とパスワード **admin** を持つ管理者です。「モニターのユーザー」1296 ページを参照してください。

   **[このコンピューターにログイン情報を記憶]** をクリックし、2 週間またはログアウトするまでセッションが維持されるようにできます。**[このコンピューターにログイン情報を記憶]** をオフにすると、ブラウザーを閉じるか、ログアウトしたときにセッションの有効期限が切れます。

### ♦ モニター Production Edition の起動 (Windows の場合)

1. Windows では、デフォルトで、モニター Production Edition のインストール時に自動的にサービスとして実行されます。サービスとして実行されているため、このモニターはコンピューターを起動すると自動的に起動します。モニターを停止した場合は、次のいずれかの手順を実行してモニターサービスを再起動できます。

   - モニターがインストールされているコンピューターで、**[スタート]** » **[プログラム]** » **[SQL Anywhere モニター 12]** » **[SQL Anywhere モニター]** をクリックします。
   - モニターのインストールディレクトリの *bin32* または *bin64* ディレクトリから *samonitor* バッチファイルを実行します。このバッチファイルによってモニターサービスが起動します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
samonitor.bat start
```

モニターはメトリックの収集を開始します。

2. モニターにログインするための URL をブラウズします。デフォルト URL は、*http://localhost:4950* です。

> **注意**
> ネットワークを経由してモニターにアクセスしている場合は、**http://***computer-name***:4950** をブラウズします。*computer-name* は、モニターが実行されているコンピューターの名前です。「モニターにリモートでログイン」1239 ページを参照してください。

3. ログインします。

   要求された場合は、モニターのユーザー名とパスワードを入力します。モニターのユーザー ID とパスワードでは、大文字と小文字が区別されます。ユーザーには、管理者、オペレーター、および読み込み専用ユーザーの 3 つの異なるタイプがあります。それぞれのユーザータイプには、異なるパーミッションが付与されています。デフォルトのユーザーは、名前 **admin** とパスワード **admin** を持つ管理者です。「モニターのユーザー」1296 ページを参照してください。

   **[このコンピューターにログイン情報を記憶]** をクリックし、2 週間またはログアウトするまでセッションが維持されるようにできます。**[このコンピューターにログイン情報を記憶]** をオフにすると、ブラウザーを閉じるか、ログアウトしたときにセッションの有効期限が切れます。

### ♦ モニター Production Edition の起動 (Linux の場合)

1. Linux では、デフォルトで、モニター Production Edition のインストール時に自動的にサービスとして実行されます。サービスとして実行されているため、このモニターはコンピューターを起動すると自動的に起動します。モニターを停止した場合は、モニターのインストールディレクトリの *bin32* または *bin64* ディレクトリから *samonitor.sh* スクリプトを実行して、モニターサービスを再起動できます。このスクリプトによってモニターサービスが起動します。

   ```
   samonitor.sh launch
   ```

   モニターはリソースに接続し、モニターに指定されているすべてのリソースのメトリックの収集を開始します。ブラウザーに、モニターにログインするためのデフォルト URL *http://localhost:4950* が表示されます。

> **注意**
> ネットワークを経由してモニターにアクセスしている場合は、**http://***computer-name***:4950** をブラウズします。*computer-name* は、モニターが実行されているコンピューターの名前です。「モニターにリモートでログイン」1239 ページを参照してください。

2. ログインします。

   要求された場合は、モニターのユーザー名とパスワードを入力します。モニターのユーザー ID とパスワードでは、大文字と小文字が区別されます。ユーザーには、管理者、オペレー

ター、および読み込み専用ユーザーの 3 つの異なるタイプがあります。それぞれのユーザータイプには、異なるパーミッションが付与されています。デフォルトのユーザーは、名前 **admin** とパスワード **admin** を持つ管理者です。「モニターのユーザー」1296 ページを参照してください。

**[このコンピューターにログイン情報を記憶]** をクリックし、2 週間またはログアウトするまでセッションが維持されるようにできます。**[このコンピューターにログイン情報を記憶]** をオフにすると、ブラウザーを閉じるか、ログアウトしたときにセッションの有効期限が切れます。

**参照**


- 「モニターの停止」1238 ページ
- 「モニター Production Edition を停止するには、次の手順に従います。」1239 ページ
- 「モニターからのログアウト」1240 ページ
- 「[概要] ダッシュボード」1240 ページ


# モニターの停止

モニターを停止すると、すべてのリソースについてメトリックの収集が停止されます。

**警告**
ほとんどの場合、モニターの実行は継続し、ブラウザーを閉じることをおすすめします。ブラウザーを閉じても、メトリックの収集は停止しません。

特定のデータベース、MobiLink サーバー、MobiLink サーバーファーム、または Relay Server ファームのモニタリングを停止する方法については、「リソースのモニタリングの停止」1258 ページを参照してください。

**♦ モニター Developer Edition の停止 (Windows の場合)**

- システムトレイで [SQL Anywhere モニター] アイコンをクリックし、**[SQL Anywhere モニターの終了]** をクリックします。

  モニターは、すべてのリソースについてメトリックの収集を停止します。

  モニターを再起動する方法については、「モニターの起動」1235 ページを参照してください。

**♦ モニター Developer Edition の停止 (Linux の場合)**

- モニターのインストールディレクトリの *bin32* または *bin64* ディレクトリから *samonitor.sh* スクリプトを実行します。

  ```
  ./samonitor.sh stop
  ```

  モニターは、すべてのリソースについてメトリックの収集を停止します。

  モニターを再起動する方法については、「モニターの起動」1235 ページを参照してください。

**モニター Production Edition を停止するには、次の手順に従います。**

モニター Production Edition は、デフォルトで、Windows または Linux へのインストール時に自動的にサービスとして実行されます。モニターおよびすべてのリソースメトリックの収集を停止するには、サービスを停止します。

**♦ モニターサービスを停止するには、次の手順に従います (Windows の場合)。**

● モニターサービスを停止するには、モニターのインストールディレクトリの *bin32* ディレクトリから *samonitor.bat* を実行します。

```
samonitor.bat stop
```

このバッチファイルによってモニターサービスが停止します。モニターは、すべてのリソースについてメトリックの収集を停止します。

モニターサービスを再起動する方法については、「モニターの起動」1235 ページを参照してください。

**♦ モニターサービスを停止するには、次の手順に従います (Linux の場合)。**

● モニターサービスを停止するには、モニターのインストールディレクトリの *bin32* または *bin64* ディレクトリから *samonitor.sh* スクリプトを実行します。

```
samonitor.sh stop
```

このスクリプトによってモニターサービスが停止します。モニターは、すべてのリソースについてメトリックの収集を停止します。

モニターサービスを再起動する方法については、「モニターの起動」1235 ページを参照してください。

**参照**


- 「モニターの起動」1235 ページ
- 「モニターにリモートでログイン」1239 ページ
- 「モニターからのログアウト」1240 ページ
- 「[概要] ダッシュボード」1240 ページ


# モニターにリモートでログイン

**♦ モニターにログインするには、次の手順に従います。**

モニターへのログインに使用するコンピューターは、モニターが実行されているネットワークに接続されている必要があります。

1. モニターがインストールされているコンピューターで、モニターを起動します。「モニターの起動」1235 ページを参照してください。

2. モニターにアクセスしているコンピューターで、モニターにログインするためのデフォルト URL **http://***computer-name***:4950** をブラウズします。*computer-name* は、モニターが実行されているコンピューターの名前です。たとえば、*http://samplehost:4950* のようになります。

3. 要求された場合は、モニターのユーザー名とパスワードを入力します。モニターのユーザー ID とパスワードでは、大文字と小文字が区別されます。「モニターのユーザー」1296 ページを参照してください。

**参照**


- 「モニターの起動」1235 ページ
- 「モニターの停止」1238 ページ
- 「モニターからのログアウト」1240 ページ
- 「[概要] ダッシュボード」1240 ページ


# モニターからのログアウト

**♦ モニターからのログアウト**

モニターからログアウトしても、メトリックの収集には影響しません。メトリックの収集を停止する場合は、次の手順に従います。

- リソースのモニタリングを停止します。「リソースのモニタリングの停止」1258 ページを参照してください。
- モニターを終了します。「モニターの停止」1238 ページを参照してください。
- 右上隅の **[ログアウト]** をクリックします。

  モニターへのログイン時に **[このコンピューターにログイン情報を記憶]** をクリックすると、ブラウザーを閉じてもモニターからログアウトしません。

**参照**


- 「モニターの起動」1235 ページ
- 「モニターの停止」1238 ページ
- 「モニターにリモートでログイン」1239 ページ
- 「[概要] ダッシュボード」1240 ページ


# [概要] ダッシュボード

**[概要]** ダッシュボードでは、モニタリングされているリソース (たとえば、SQL Anywhere データベース、Mobile Link サーバーおよび Mobile Link サーバーファーム、または Relay Server ファーム) の正常性や可用性の概要が表示されます。

デフォルトで、**[概要]** ダッシュボードには、**[リソースリスト]** ウィジェットと **[警告リスト]** ウィジェットがあります。

## [リソースリスト] ウィジェット

**[リソースリスト]** ウィジェットには、モニタリングされているリソースがリストされた表が含まれ、各リソースの全体的な正常性に関する表示が提供されます。表には常に **[SQL Anywhere モニター]** という名前のデフォルトリソースがあります。このリソースは、モニター自体の正常性についてレポートします。**[SQL Anywhere モニター]** リソースを変更したり、このリソースのモニタリングを停止したりできません。

**♦ リソースの詳細を表示するには、次の手順に従います。**

● **[リソースリスト]** ウィジェットで、リソース名をクリックします。

選択したリソースのダッシュボードが開きます。「ダッシュボード」1243 ページを参照してください。

**[リソースリスト]** ウィジェットで、**[ステータス]** カラムには、モニターとそのリソース間の接続に関する情報が示されます。**[ステータス]** カラムには、リソースに対していずれかのユーザーによる何らかの操作が必要かどうかが示されます。

リソースのモニターの状態は次のいずれかになります。

| アイコン | ステータス | 説明 |
|---|---|---|
| アイコンなし | [正常] | リソースに未解決の警告はありません。 |
| ⚠ | [警告あり] | リソースに 1 つ以上の未解決の警告があります。 |
| ✖ | [使用不可] | リソースは使用できません。たとえば、リソースはダウンしています。 |

| アイコン | ステータス | 説明 |
|---|---|---|
| | [モニター停止] | ブラックアウトのため、またはユーザーがリソースのモニタリングを手動で停止したため、リソースはモニタリングされていません。 |

### [警告リスト] ウィジェット

[警告リスト] ウィジェットには、モニタリング対象のデータベース、Mobile Link サーバーおよび Mobile Link サーバーファーム、および Relay Server ファームに関する警告があります。

**♦ 警告の詳細を表示するには、次の手順に従います。**

● [警告リスト] で警告をクリックし、[詳細] をクリックします。

警告の詳細が表示されたウィンドウが開きます。

警告の状態は次のいずれかになります。

| アイコン | ステータス | 説明 |
|---|---|---|
| | [アクティブ] | 警告の条件は引き続き適用されます。警告は解決されていません。 |
| | [非アクティブ] | 警告のトリガーとなった問題は存在しなくなりましたが、警告は解決または削除されていません。 |
| | [解決済み] | 管理者またはオペレーターがその警告に解決済みのマークを付けました。 |

警告の重大度のレベルは次のいずれかになります。

| アイコン | 重大度 | 説明 |
|---|---|---|
| | 重大度高 | 重大度高の警告は、ユーザーがすぐに注意を向ける必要がある問題を示します。たとえば、リソースがディスク領域の低スレッショルド値を超えた場合、重大度高の警告が発行されます。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| アイコン | 重大度 | 説明 |
|---|---|---|
| ◐ | 重大度中 | 重大度中の警告は、拡大する可能性があるため、ユーザーが注意を向ける必要がある問題を示します。たとえば、リソースが CPU 使用率のスレッショルドを超えた場合、重大度中の警告が発行されます。 |
| ○ | 重大度低 | 重大度低は問題を示します。たとえば、リソースが接続に失敗した場合、重大度低の警告が発行されます。 |

**参照**

- 「ウィジェット」1244 ページ
- 「警告」1301 ページ

## ダッシュボード

ダッシュボードは、各ユーザーに固有です。どのユーザーも各自のダッシュボードを追加、編集、または削除できます。

**♦ 新しいダッシュボードを追加するには、次の手順に従います。**

1. モニターにログインします。
2. **[ダッシュボード]** ウィンドウ枠で、**[新規追加]** をクリックします。
3. ウィンドウの指示に従ってダッシュボードを追加します。
4. **[OK]** をクリックします。

   モニターでは新しいダッシュボードを作成し、開きます。左のナビゲーションメニューの **[ダッシュボード]** リストにダッシュボードの名前も表示されます。

**♦ ダッシュボードを編集または削除するには、次の手順に従います。**

1. ダッシュボードを開きます。
2. ダッシュボードの右上で、**[カスタマイズ]** をクリックし、次のいずれかをクリックします。
   - **[設定]**　ダッシュボードの設定を編集します。
   - **[削除]**　ダッシュボードを削除します。

### ダッシュボードテンプレート

ダッシュボードが作成されるときに、デフォルトのウィジェットが取り込まれます。指定したダッシュボードのウィジェットとレイアウトを使用するようにモニターを設定することによって、新しいダッシュボードに表示されるデフォルトのウィジェットを変更できます。「ウィジェット」1244 ページを参照してください。

♦ **ダッシュボードテンプレートを作成するには、次の手順に従います。**

1. テンプレートとして使用するダッシュボードを開きます。
2. ダッシュボードの右上で、**[カスタマイズ]** をクリックし、**[テンプレートとして設定]** をクリックします。

   新しいダッシュボードを追加すると、ダッシュボードにはテンプレートダッシュボードと同じウィジェットとレイアウトがあります。

## ウィジェット

どのユーザーも各自のダッシュボードにあるウィジェットを追加、編集、または削除できます。ダッシュボードが作成されるときに、デフォルトのウィジェットが取り込まれます。

次の表には、リソースに対して作成可能なさまざまなタイプのメトリックウィジェットが含まれています。

| メトリックのタイプ | 説明 |
|---|---|
| メトリック | モニタリングしているリソースのために収集された情報を表示します。メトリックは、グラフと表のどちらでも表示を選択できます。「メトリックのリスト」1265 ページを参照してください。 |
| 警告 | リソースに対する警告のリストを表示します。[警告リスト] ウィジェット 1242 ページを参照してください。 |
| リソース | リソースのリストを表示します。[リソースリスト] ウィジェット 1241 ページを参照してください。 |
| **SQL Anywhere** の接続 | SQL Anywhere データベース接続のリストを表示します。「モニターを使用した接続の切断」1247 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| メトリックのタイプ | 説明 |
| --- | --- |
| **[SQL Anywhere スケールアウトトポロジ]** | SQL Anywhere のミラーリングおよびスケールアウトシステムのトポロジを表示します。「[SQL Anywhere スケールアウトトポロジ] ウィジェット」1245 ページと「レッスン 3：SQL Anywhere モニターからの読み込み専用スケールアウトシステムのモニタリング」1193 ページを参照してください。 |

**♦ ウィジェットを追加するには、次の手順に従います。**

1. ダッシュボードを開きます。
2. ダッシュボードの右上で、**[カスタマイズ]** をクリックし、**[ウィジェットの追加]** をクリックします。
3. **[必要なウィジェットのタイプを指定してください。]** フィールドで必要なウィジェットのタイプをクリックしてから、**[次へ]** をクリックします。
4. **[このウィジェットの名前を指定してください。]** フィールドでウィジェットの名前を指定します。
5. **[どのようなリソースに関心がありますか ?]** フィールドでリソースを指定します。
6. **[ウィジェットの追加]** ウィンドウの指示に従ってウィジェットを追加します。
7. **[作成]** をクリックします。

**♦ ウィジェットを編集または削除するには、次の手順に従います。**

- ウィジェットウィンドウ枠で、ウィジェットの右上にあるドロップダウン矢印をクリックし、次のいずれかをクリックします。
  - **[設定]**　ウィジェットの設定を編集します。
  - **[削除]**　ウィジェットを削除します。

**参照**

- 「ダッシュボード」1243 ページ
- 「[概要] ダッシュボード」1240 ページ

## [SQL Anywhere スケールアウトトポロジ] ウィジェット

**[SQL Anywhere スケールアウトトポロジ]** ウィジェットを作成することで、SQL Anywhere スケールアウトおよびミラーリングシステムのトポロジを表示できます。

### トポロジウィジェットでの読み込み専用スケールアウトシステムの表示

読み込み専用スケールアウトシステム用の **[SQL Anywhere スケールアウトトポロジ]** ウィジェットは、次のフィールドとその可能な値を表示します。

- **[サーバー名]** スケールアウトシステムのノード名。
- **[タイプ]** サーバーのロール。タイプは、読み込み専用スケールアウトシステムが作成されたときにサーバーが定義された方法によって、**[partner - primary]** または **[コピー]** のいずれかになります。
- **[状態]** ノードの接続状況。**[接続しました]** または **[切断しました]** のいずれか 1 つになります。
- **[接続]** このノードへの現在の接続数。
- **[最終更新]** サーバーがステータスを最後に更新した時刻。

### トポロジウィジェットでのデータベースミラーリングシステムの表示

データベースミラーリングシステム用の **[SQL Anywhere スケールアウトトポロジ]** ウィジェットは、次のフィールドとその可能な値を表示します。

- **[サーバー名]** ミラーリングシステムにデータベースサーバーの名前を表示します。
- **[タイプ]** サーバーのロールが一覧表示され、次の値の 1 つが表示されます。

  **[partner - primary]** この値によって、現在プライマリとして動作しているサーバーが識別されます。

  **[partner - mirror]** この値によって、現在ミラーとして動作しているサーバーが識別されます。

  **[arbiter]** この値によって、監視サーバーが識別されます。
- **[状態]** 各サーバーの接続状況が表示されます。
- **[接続]** サーバーへの現在の接続数が表示されます。
- **[最終更新]** サーバーのステータスが更新された最新の時刻が表示されます。

### 参照


- 「CREATE MIRROR SERVER 文」『SQL Anywhere サーバー SQL リファレンス』
- 「sa_mirror_server_status システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』
- 「データベースミラーリング」1137 ページ
- 「SQL Anywhere の読み込み専用のスケールアウト」1188 ページ
- 「SQL Anywhere モニター」1217 ページ
- 「チュートリアル：モニターによるリソースのモニタリング」1222 ページ
- 「チュートリアル：読み込み専用のスケールアウトシステムの作成とモニタリング」1190 ページ
- 「チュートリアル：データベースミラーリングの使用」1146 ページ

## モニターを使用した接続の切断

♦ **接続を切断するには、次の手順に従います (モニターの場合)。**

データベースの DBA ユーザー名とパスワードを知っている場合、どのユーザーもモニターを使用してデータベースリソースへの接続を表示および切断できます。

1. モニターにログインします。
2. SQL Anywhere リソースのダッシュボードを開きます。

   **[ダッシュボード]** » *ダッシュボードの名前*をクリックします。
3. **[接続]** ウィジェットで、接続を選択し、接続名の横にある **x** をクリックします。
4. 要求された場合は、データベースの DBA ユーザー名とパスワードを入力します。

   接続を切断する場合は、データベースの DBA ユーザー ID とパスワードを指定する必要があります。このクレデンシャルは、データベースとの接続および指定した接続の切断に使用されます。これにより、DBA クレデンシャルがモニターから削除されます。
5. **[OK]** をクリックします。

## 時刻の表示方法の概要

♦ **モニターとブラウザー間の時刻の差異を確認するには、次の手順に従います。**

モニターで表示されるすべての時刻は、24 時間表示に基づいており、モニターが実行されているコンピューターのローカル時間に対応しています。

1. モニターにログインします。
2. リソースのダッシュボードを開きます。
3. **[サーバー情報]** ウィジェットで、**[サーバー時間オフセット]** メトリックを確認します。

   **[サーバー時間オフセット]** には、モニターが実行されているコンピューターの時刻と、モニターデータの表示に使用しているコンピューターの時刻の差異が記録されます。

**参照**

- 「メトリック」1262 ページ

# [管理] ウィンドウ

> **注意**
> **[管理]** ウィンドウにアクセスできるのは管理者だけです。管理者の詳細については、「モニターのユーザー」1296 ページを参照してください。

管理者として、モニタリングするリソース (たとえば、SQL Anywhere データベース、Mobile Link サーバーおよび Mobile Link サーバーファームまたは Relay Server ファーム) を選択し、次の操作を行うことができます。

●リソースの追加、編集、削除。「リソース」1249 ページを参照してください。

●ユーザーの追加および編集。「モニターのユーザー」1296 ページを参照してください。

●モニターの設定。「モニターのバックアップ」1309 ページを参照してください。

●メッセージログの表示。「メッセージログの表示。」1248 ページを参照してください。

●例外レポートの表示。「例外レポートの表示」1248 ページを参照してください。

**参照**

● 「メトリック」1262 ページ

## メッセージログの表示。

**♦ メッセージログを表示するには、次の手順に従います。**

**[メッセージログ]** には、SQL Anywhere モニターの動作に関する SQL Anywhere モニターからの情報メッセージが表示されます。メッセージは、テーブル内に最新のメッセージを一番上にして表示されます。

1. モニターに管理者としてログインします。
2. **[ツール]** » **[管理]** をクリックします。
3. **[メッセージログ]** をクリックします。

## 例外レポートの表示

**♦ 例外レポートを表示するには、次の手順に従います。**

SQL Anywhere モニターで致命的なエラーやクラッシュが発生すると、問題が発生したときの状況に関する例外レポートが作成されます。

1. モニターに管理者としてログインします。
2. **[ツール]** » **[管理]** をクリックします。
3. **[例外レポート]** をクリックします。

# リソース

「リソース」とは、SQL Anywhere データベース (読み込み専用スケールアウトシステムまたはミラーリングシステムのデータベースを含む)、Mobile Link サーバー、Mobile Link サーバーファーム、または Relay Server ファームのことです。管理者は、リソースをモニターに追加し、モニタリングを開始できます。

### SQL Anywhere モニターリソース

**[SQL Anywhere モニター]** という名前のデフォルトリソースは、モニター自体の正常性についてレポートします。このデフォルトリソースは、モニターが実行されているコンピューターをモニタリングし、モニターに問題が発生したときに警告を送信する場合に役立ちます。このリソースを変更したり、このリソースのモニタリングを停止することはできません。

### 参照

- [リソースリスト] ウィジェット 1241 ページ

# リソースの追加

データベース (読み込み専用スケールアウトシステムまたはミラーリングシステムのデータベースを含む) Mobile Link サーバー、Mobile Link サーバーファーム、または Relay Server ファームをモニタリングするには、管理者が目的のリソースをあらかじめモニターに追加しておく必要があります。デフォルトでは、リソースを追加した後、モニタリングが開始されます。

管理者は、一度に 1 つずつリソースを追加したり、インポートファイルを使用して複数のリソースを追加したりできます。「複数のリソースの追加」1252 ページを参照してください。

**♦ データベース Mobile Link サーバー Relay Server ファームリソースをモニターに追加するには、次の手順に従います。**

1. モニターに管理者としてログインします。
2. **[管理]** をクリックします。
3. **[リソース]** をクリックします。
4. **[追加]** をクリックします。
5. **[リソースの追加]** ウィンドウで、次の手順に従います。
   a. モニタリングするリソースのタイプをクリックします。読み込み専用スケールアウトシステムまたはミラーリングシステムのデータベースをモニタリングするには、**[SQL Anywhere データベース]** リソースを追加します。
   b. **[次へ]** をクリックします。
   c. **[名前]** フィールドにリソースの名前を入力します。これはモニターに表示される名前です。次に **[次へ]** をクリックします。

d. **[ホスト]** フィールドで、データベースサーバー、MobiLink サーバー、または Relay Server ファームが実行されているコンピューターのホスト名または IP アドレスを指定します。現在のコンピューターを表す localhost という名前を使用できます。この値は必須です。「Host 接続パラメーター」346 ページを参照してください。

読み込み専用スケールアウトシステムとその他のデータベースサーバーで、ルートノード用のリソース設定には違いはありません。リソースの設定時や移行時に、モニターはリソースが読み込み専用スケールアウトシステムのルートノードであることを検出します。

ミラーリングシステムのプライマリデータベースのリソースには、特別な設定が必要です。フェールオーバーが発生してもモニターが常にプライマリデータベースサーバーをモニタリングするようにするには、次のオプションを使用してリソースを設定する必要があります。

- **[ホスト]** フィールドでプライマリサーバとミラーサーバーのホスト名とポート番号をカンマ区切りのリストで指定します。(例：my-primary-server:2638,my-mirror-server:49152)。
- **[ポート]** フィールドで、フィールドが空であることを確認します。

  **[サーバー]** フィールドにプライマリサーバーの代替サーバー名を入力します。つまり、データベースミラーリングシステムのプライマリサーバーとして動作するデータベースサーバーに接続するためにクライアントで使用する名前を指定します。

e. **[次へ]** をクリックします。

f. **[リソースの追加]** ウィンドウの指示に従って、データベース、Mobile Link サーバー、または Relay Server ファームをモニタリングするためのリソースを追加します。

MobiLink でリソースを追加する場合は、Mobile Link サーバーに接続するための Mobile Link ユーザー ID とパスワードを指定する必要があります。これらのクレデンシャルは、Mobile Link サーバーへの接続に使用されます。Mobile Link ユーザー ID とパスワードはモニターに格納されます。

g. **[作成]** をクリックします。

リソースが追加され、モニタリングが開始されます。

6. データベースの場合は、要求されたら、データベースの DBA ユーザー ID とパスワードを入力し、**[閉じる]** をクリックします。

これらのクレデンシャルは次の操作を行う場合に使用されます。

- データベースに接続します。
- sa_monitor_user という名前の新しいユーザーを作成します。モニターは、sa_monitor_user を使用してデータベースに接続し、モニタリングします。
- データベースをモニタリングするために sa_monitor_user が必要とするデータベースオブジェクトをインストールします。 インストールされるオブジェクトのリストについては、「モニターがインストールしたデータベースオブジェクトのリスト」1311 ページを参照してください。
- モニターから DBA クレデンシャルを破棄します。sa_monitor_user はモニタリングされるデータベースに追加されるので、そのデータベース外部のどこかに DBA クレデンシャル

を格納する必要はありません。「モニターがインストールしたデータベースオブジェクトのリスト」1311 ページを参照してください。

リソースが追加され、モニタリングが自動的に開始されます。

7. **[閉じる]** をクリックします。
8. **[閉じる]** をクリックします。

   **[概要]** ダッシュボードの **[リソースリスト]** にリソースが表示されます。

9. (任意) リソースのダッシュボードを追加します。デフォルトでは、リソースの追加時にダッシュボードは追加されません。
   a. **[概要]** ダッシュボードを開きます。

      **[ダッシュボード]** » **[概要]** をクリックします。

   b. **[リソースリスト]** で新しいリソースをクリックします。

      モニターではリソースの新しいダッシュボードを作成し、開きます。新しいダッシュボードの名前がサイドバーの **[ダッシュボード]** セクションに表示されます。「ダッシュボード」1243 ページを参照してください。

10. (任意) ダッシュボードにウィジェットを追加します。「ウィジェット」1244 ページを参照してください。

    SQL Anywhere 読み込み専用スケールアウトシステムおよびミラーリングシステムのトポロジを表示するには、**[SQL Anywhere スケールアウトトポロジ]** ウィジェットを追加する必要があります。「[SQL Anywhere スケールアウトトポロジ] ウィジェット」1245 ページを参照してください。

**♦ Mobile Link サーバーファームリソースをモニターに追加するには、次の手順に従います。**

1. モニターに管理者としてログインします。
2. **[管理]** をクリックします。
3. **[リソース]** をクリックします。
4. [追加] をクリックします。
5. **[リソースの追加]** ウィンドウの指示に従って、Mobile Link サーバーファームをモニタリングするためのリソースを追加します。

   モニターは、Mobile Link サーバーファームおよびファームを構成する各サーバーについてのメトリックを収集します。Mobile Link サーバーファームをリソースとして追加する場合は、ファームを構成するサーバーを指定し、必要に応じてサーバーをリソースとして追加します。

6. **[作成]** をクリックします。

   リソースが追加され、モニタリングが開始されます。

7. **[閉じる]** をクリックします。

8. **[閉じる]** をクリックします。

   **[概要]** ダッシュボードの **[リソースリスト]** にリソースが表示されます。

**参照**

- 「複数のリソースの追加」1252 ページ
- 「モニターの起動」1235 ページ
- 「Mobile Link ユーザー」『Mobile Link クライアント管理』

## 複数のリソースの追加

**♦ 複数のリソースをモニターに追加するには、次の手順に従います。**

管理者は、値をカンマで区切った (CSV) ファイルを作成し、ファイルをインポートすることによって、モニターに複数のリソースを追加できます。

データベースミラーリングシステムにリソースとしてプライマリデータベースを追加するときに、CSV ファイルは使用できません。その代わりに、**[管理]** ウィンドウを使用します。「リソースの追加」1249 ページを参照してください。

1. CSV ファイルを作成します。

   CSV ファイルの各行には、単一のリソースについての情報が含まれています。行内のカンマで区切られた各項目は、リソースの属性を記述します。項目の順序が重要です。次の表に、項目とその順序を説明します。

   オプションの項目の場合、その前にある項目 (他のオプションの項目も含む) を指定する必要があります。たとえば、データベース名を指定する場合、サーバー名とポート番号も指定する必要があります。暗号化タイプを指定する場合、接続タイプも指定する必要があります。モニタリングの URL サフィックスを指定する場合、ポート番号も指定する必要があります。

   - **SQL Anywhere データベースリソースの CSV ファイルフォーマット**　オプションの項目の場合、その前にある項目 (他のオプションの項目も含む) を指定する必要があります。たとえば、データベース名を指定する場合、サーバー名とポート番号も指定する必要があります。

| 行内の位置 | 項目 | 説明 |
|---|---|---|
| 1 | リソースタイプ | **sa** と入力してリソースが SQL Anywhere データベースであることを指定します。この項目は必須です。 |
| 2 | リソース名 | モニターでデータベースに使用するリソース名を指定します。この項目は必須です。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 行内の位置 | 項目 | 説明 |
|---|---|---|
| 3 | ユーザー名 | データベースへの接続に使用する DBA ユーザー名を指定します。この項目は必須です。 |
| 4 | パスワード | データベースに接続するための DBA パスワードを指定します。この項目は必須です。 |
| 5 | ホスト (SQL Anywhere) | データベースサーバーを実行するコンピューターのホスト名または IP アドレスを指定します。現在のコンピューターを表す localhost という名前を使用できます。この項目は必須です。 |
| 6 | ポート (SQL Anywhere) | SQL Anywhere データベースサーバーが実行されているポート番号を指定します。SQL Anywhere のデフォルトポート番号は 2638 です。この項目はオプションです。デフォルト値は 0 です。 |
| 7 | サーバー (SQL Anywhere) | データベースが接続されるデータベースサーバーの名前を指定します。この項目はオプションです。 |
| 8 | データベース (SQL Anywhere) | データベースの名前を指定します。この項目はオプションです。 |
| 9 | 接続パラメーター (SQL Anywhere) | データベースへの接続時に使用する追加の接続パラメーターを指定します。option=value の組み合わせをセミコロンで区切ったリストで、接続パラメーターを取得します。デフォルト値は "" です。この項目はオプションです。 |

次に、2 つの SQL Anywhere データベースリソースを含む csv ファイルの例を示します。

```
sa,demo,DBA,sql,localhost,2638,demo12
sa,demo3,DBA,sql,localhost,49152,demo12,demo
```

- **Mobile Link リソースの CSV ファイルフォーマット**　オプションの項目の場合、その前にある項目 (他のオプションの項目も含む) を指定する必要があります。たとえば、暗号化タイプを指定する場合、接続タイプも指定する必要があります。

| 行内の位置 | 項目名 | 説明 |
|---|---|---|
| 1 | リソースタイプ | **ml** と入力してリソースが Mobile Link サーバーであることを指定します。<br><br>この項目は必須です。 |
| 2 | リソース名 | モニターでの表示に使用するリソース名を指定します。この項目は必須です。 |
| 3 | ユーザー名 | リソースに接続するためのユーザー名を指定します。この項目は必須です。 |
| 4 | パスワード | リソースに接続するためのパスワードを指定します。この項目は必須です。 |
| 5 | ホスト (Mobile Link サーバー) | サーバーが稼働しているコンピューターのホスト名または IP アドレスを指定します。現在のコンピューターを表す localhost という名前を使用できます。この項目は必須です。 |
| 6 | ポート (Mobile Link サーバー) | Mobile Link サーバーが実行されているポート番号を指定します。Mobile Link のデフォルトポート番号は 2439 です。この項目はオプションです。デフォルト値は 0 です。 |


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 行内の位置 | 項目名 | 説明 |
|---|---|---|
| 7 | 接続タイプ (Mobile Link サーバー) | リソースへの接続に使用するメソッドを指定します。次のいずれかを入力します。<br>○**tcpip**<br>これはデフォルトです。<br>○**http**<br>○**https**<br>○**tls**<br>この項目はオプションです。 |
| 8 | 暗号化タイプ (Mobile Link サーバー) | 接続の暗号化に使用するメソッドを指定します。次のいずれかを入力します。<br>○ **N** 暗号化なし。これはデフォルトです。<br>○ **E** ECC 暗号化。<br>○ **R** RSA 暗号化。<br>○ **F** FIPS 認定 RSA 暗号化<br>この項目はオプションです。 |
| 9 | 接続パラメーター (Mobile Link サーバー) | データベースへの接続時に使用する追加の接続パラメーターを指定します。option=value の組み合わせをセミコロンで区切ったリストで、接続パラメーターを取得します。この項目はオプションです。 |

Mobile Link サーバー：

次に、2 つの Mobile Link サーバーリソースを含むインポートファイルの例を示します。

```
ml,demo2,DBA,sql,localhost
ml,demo4,DBA,sql,localhost,0,tcpip,N
```

● **Mobile Link サーバーファームリソースの CSV ファイルフォーマット**

| 行内の位置 | 項目名 | 説明 |
| --- | --- | --- |
| 1 | リソースタイプ | **mf** と入力してリソースが Mobile Link サーバーファームであることを指定します。<br><br>この項目は必須です。 |
| 2 | リソース名 | モニターでの表示に使用するリソース名を指定します。この項目は必須です。 |
| 3 | Mobile Link サーバー名 | Mobile Link サーバーファームに含める Mobile Link サーバーリソースを指定します。Mobile Link サーバーリソースをカンマで区切られたリストとしてリストします。この項目は必須です。 |

Mobile Link サーバーファーム：

次に、2 つの Mobile Link サーバーリソースと 1 つの Mobile Link サーバーファームリソースを含むインポートファイルの例を示します。

```
ml,demo2,DBA,sql,localhost
ml,demo4,DBA,sql,localhost,0,tcpip,N
mf,collection_demo,demo2,demo4
```

- **Relay Server ファームリソースの CSV ファイルフォーマット**　オプションの項目の場合、その前にある項目 (他のオプションの項目も含む) を指定する必要があります。たとえば、Monitoring URL サフィックスを指定する場合、ポート番号も指定する必要があります。

| 行内の位置 | 項目名 | 説明 |
| --- | --- | --- |
| 1 | リソースタイプ | **rf** と入力してリソースが Relay Server ファームであることを指定します。<br><br>この項目は必須です。 |
| 2 | リソース名 | モニターでの表示に使用するリソース名を指定します。この項目は必須です。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 行内の位置 | 項目名 | 説明 |
|---|---|---|
| 3 | ホスト (Relay Server ファーム) | サーバーが稼働しているコンピューターのホスト名または IP アドレスを指定します。現在のコンピューターを表す localhost という名前を使用できます。この項目は必須です。 |
| 4 | ポート (Relay Server ファーム) | Relay Server ファームが実行されているポート番号を指定します。この項目はオプションです。デフォルト値は 0 です。 |
| 5 | モニタリングの URL サフィックス | Relay Server ファームのモニタリングエージェントを特定するための URL サフィックスを指定します。この項目はオプションです。 |
| 6 | HTTPS | Relay Server ファームへの接続に HTTPS を使用するようにモニターに **True** を指定します。この項目はオプションです。<br><br>デフォルトは **False** です。この項目はオプションです。 |
| 7 | HTTP 認証用のユーザー名 | 必要に応じて、HTTPS 認証ユーザー名を指定します。この項目はオプションです。 |
| 8 | HTTP 認証用のパスワード | 必要に応じて、HTTPS 認証パスワードを指定します。この項目はオプションです。 |

次に、2 つの Mobile Link サーバーリソース、1 つの Mobile Link サーバーファームリソース、1 つの Relay Server ファームリソースを含むインポートファイルの例を示します。

```
rf,rf_demo,rstest02.website.com
```

2. モニターに管理者としてログインします。
3. **[管理]** をクリックします。
4. **[リソース]** をクリックします。
5. **[インポート]** をクリックします。
6. CSV ファイルを検索し、**[開く]** をクリックします。

CSV ファイルのリソースがモニターに追加され、モニタリングが開始されます。

7. **[閉じる]** をクリックします。

8. **[閉じる]** をクリックします。

9. **[概要]** ダッシュボードの **[リソースリスト]** にインポートされたリソースが追加されます。

## リソースのモニターリングの開始

リソースのモニタリングは、次の場合に開始されます。

- モニターの起動時 (自動的に開始)。モニターを起動すると、デフォルトで、モニターはリソースに接続し、指定されているすべてのリソースのメトリックを収集します。

- 管理者がリソースを追加したとき (自動的に開始)。リソースが追加されると、モニターは、リソースへの接続とモニタリングの開始を試みます。「リソースの追加」1249 ページを参照してください。

- ブラックアウト期間が終了したとき (自動的に開始)。モニターは、リソースへの接続とモニタリングの再開を自動的に試みます。

- 管理者が、**[管理]** ウィンドウを開き、**[リソース]** をクリックし、リストからリソースを選択し、**[起動]** をクリックしたとき。

## リソースのモニタリングの停止

モニターでメトリックを収集しない場合は、リソースのモニタリングを停止する必要があります。たとえば、目的のリソースが使用不可のときにモニタリングを停止します。そうしないと、リソースが使用可能になるまで警告を受信し続けることになります。デフォルトリソースであるモニターを除き、リソースのモニタリングはいつでも停止できます。

リソースのモニタリングを停止すると、モニターは次のように動作します。

- 該当するリソースのメトリックの収集を停止します。

- 該当するリソースに関する警告の発行を停止します。

リソースのモニタリングを停止するには、次の 2 つの方法があります。

- **定期的に繰り返されるブラックアウト期間をスケジュールする**　この方法は、次の状況が当てはまる場合に適しています。

  - リソースのモニターリングを繰り返し停止する必要がある場合。たとえば、毎月末に定期メンテナンスを行う場合です。

  - リソースが使用不可になる期間があらかじめわかっている場合。たとえば、定期メンテナンスに 4 時間かかることがわかっている場合です。

○モニタリングを自動的に再開する必要がある場合。ブラックアウト期間が終わると、モニターはリソースへの再接続を行い、データの収集を続けます。

この方法を使用するには、ブラックアウト期間を設定して、指定した時刻にモニターによるモニタリングが停止されるようにする必要があります。「ブラックアウトの設定」1260 ページを参照してください。

● **モニタリングの手動停止**　この方法は、次の状況が当てはまる場合に適しています。

○まれにしか発生しないタスクや 1 回限りのタスクのためにモニタリングを停止する必要がある場合。たとえば、リソースが実行されているコンピューターを特別なメンテナンスのためにオフラインにする必要があり、モニタリングを停止する必要がある場合です。

○モニタリングを後で再開するときに立ち会える場合。リソースが手動で停止されている場合、モニターはモニタリングの再開を待機します。

この方法を使用する場合については、「リソースのモニタリングの手動停止」1259 ページを参照してください。

リソースのモニタリングを永久に停止する場合は、該当するリソースをモニターから削除することも可能です。「リソースの削除」1261 ページを参照してください。

## リソースのモニタリングの手動停止

♦ **リソースのモニタリングを手動で停止するには、次の手順に従います。**

リソースを停止したときの動作の詳細については、「リソースのモニタリングの停止」1258 ページを参照してください。

1. モニターに管理者としてログインします。
2. **[ツール]** » **[管理]** をクリックします。
3. **[リソース]** をクリックします。
4. 目的のリソースをクリックし、**[停止]** をクリックします。

> **注意**
> Mobile Link サーバーファームのモニタリングを停止すると、ファームを構成する各 Mobile Link サーバーのモニタリングは停止します。

5. **[閉じる]** をクリックします。

**参照**

- 「リソースの追加」1249 ページ
- 「ブラックアウトの設定」1260 ページ

## ブラックアウトの設定

♦ **ブラックアウト期間を設定するには、次の手順に従います。**

ブラックアウトとは、モニターでメトリックを収集しない期間のことです。ブラックアウト期間が終わると、モニターはリソースへの再接続を行い、データの収集を続けます。

リソースを停止したときの動作、およびブラックアウトの使用に適した状況の詳細については、「リソースのモニタリングの停止」1258 ページを参照してください。

ブラックアウト期間は、該当リソースのローカル時間に対応します。「時刻の表示方法の概要」1247 ページを参照してください。

1. モニターに管理者としてログインします。
2. **[ツール]** » **[管理]** をクリックします。
3. **[リソース]** をクリックします。
4. 目的のリソースをクリックし、**[設定]** をクリックします。
5. **[ブラックアウト]** をクリックします。
6. **[新規]** をクリックします。
7. **[新規ブラックアウト期間]** ウィンドウで、ブラックアウトの日付と時刻 (24 時間表記) を指定します。

   この日時は、対象リソースであるデータベース、Mobile Link サーバー、または Relay Server ファームがあるコンピューターのローカル時間に対応します。
8. **[保存]** をクリックします。
9. **[保存]** をクリックします。
10. **[閉じる]** をクリックします。
11. **[閉じる]** をクリックします。

**参照**


- 「リソースの追加」1249 ページ
- 「リソースのモニタリングの手動停止」1259 ページ


# データベースリソースの修復

♦ **SQL Anywhere リソースを修復するには、次の手順に従います。**

リソースを修復すると、該当リソースのモニタリングに必要なデータベースオブジェクトが再インストールされます。モニタリングオプションは変更されません。 インストールされるオブ

ジェクトのリストについては、「モニターがインストールしたデータベースオブジェクトのリスト」1311 ページを参照してください。

管理者は、SQL Anywhere データベースリソースのみを修復できます。デフォルトリソースである **SQL Anywhere モニター** という名前のモニターを修復することはできません。リソースを修復するたびに、データベースの DBA ユーザー ID とパスワードを指定する必要があります。

1. モニターに管理者としてログインします。
2. **[管理]** をクリックします。
3. **[リソース]** をクリックします。
4. 修復するデータベースリソースをクリックします。
5. 目的のリソースが現在モニタリングされている場合は、**[停止]** をクリックします。
6. **[修復]** をクリックします。
7. 要求されたら、SQL Anywhere データベースの DBA ユーザー ID とパスワードを入力します。この DBA クレデンシャルはデータベースとの接続に使用され、後でモニターから破棄されます。
8. **[修復]** をクリックします。
9. **[OK]** をクリックします。
10. **[開始]** をクリックして、リソースのモニタリングを再開します。「リソース」1249 ページを参照してください。
11. **[閉じる]** をクリックします。

## リソースの削除

**♦ リソースを削除するには、次の手順に従います。**

削除するリソースは、使用されなくなったデータベース Mobile Link サーバーまたは Mobile Link サーバーファーム Relay Server ファームなど、モニタリングが不要になったことが確実なリソースに限定してください。

リソースを削除したときのモニターの動作は次のとおりです。

- 該当リソースのモニタリングが永久的に停止されます。
- 該当リソースについて収集されたメトリックが破棄されます。

データベースリソースを削除しても、データベースにインストールされたモニタリングオブジェクトは削除されません。これらのオブジェクトの削除の詳細については、「モニタリングオブジェクトの削除」1312 ページを参照してください。

リソースを削除できるのは管理者だけです。**SQL Anywhere モニター**リソースは削除できません。

1. モニターに管理者としてログインします。
2. **[管理]** をクリックします。
3. **[リソース]** をクリックします。
4. リソースをクリックします。
5. **[削除]** をクリックします。
6. **[はい]** をクリックします。

> **注意**
> Mobile Link サーバーファームを削除すると、ファームを構成する Mobile Link サーバーを削除するよう要求されます。指定された Mobile Link サーバーを削除する場合は **[はい]** をクリックし、Mobile Link サーバーファームのみを削除し、指定された Mobile Link サーバーファームを削除しない場合は、**[いいえ]** をクリックします。

7. **[閉じる]** をクリックします。

**参照**

- 「リソースのモニタリングの停止」1258 ページ

# メトリック

モニターがデータベース、Mobile Link サーバー、Mobile Link サーバーファーム、および Relay Server ファームから収集し、保存するメトリックには次のものが含まれますが、これに限定されません。

- リソースが実行中がどうか。
- リソースを実行しているコンピューターが、正常に実行されているかどうか、およびネットワークに接続されているかどうか。
- リソースが要求を受信して処理しているかどうか。
- 実行時間が長いクエリ、ブロックされたユーザーなど、明らかな問題がないかどうか。
- 所定の期間内に Mobile Link サーバーが実行した同期の回数。
- バックエンドサーバーが HTTP 要求の処理に要した平均時間。

### 収集間隔

モニターに表示されるメトリックの正確さは、収集間隔によって決まります。管理者は、モニターによってメトリックが収集される頻度を変更できます。デフォルトでは、メトリックは 30 秒ごとに収集されます。

**♦ リソースの収集間隔を設定するには、次の手順に従います。**

1. モニターに管理者としてログインします。
2. **[管理]** をクリックします。
3. **[リソース]** をクリックします。
4. 目的のリソースをクリックし、**[設定]** をクリックします。
5. **[収集間隔]** をクリックします。
6. メトリックが収集される間隔を指定します (*時間* : *分* : *秒*)。収集間隔は 10 秒以上にします。
7. **[保存]** をクリックします。
8. ユーザー ID とパスワードを入力して、**[OK]** をクリックします。
9. **[閉じる]** をクリックします。
10. **[閉じる]** をクリックします。

### 参照

● 「メトリックのリフレッシュ」 1264 ページ

## メトリックのエクスポート

**♦ メトリックのエクスポート**

グラフや表を関連付けられたメトリックを XML ファイルにエクスポートできます。たとえば、**[主要なパフォーマンスのメトリック]** ウィジェットのほとんどのメトリックをエクスポートできます。ファイルにエクスポートできるデータの最大量は 25 メトリックまたは 100 万ポイントです。

1. モニターにログインします。
2. ダッシュボードを開きます。
3. メトリックが表示されているウィジェットで、ドロップダウンメニューの矢印をクリックし、**[エクスポート]** をクリックします。
4. **[メトリックのエクスポート]** ウィンドウの指示に従ってメトリックをエクスポートします。

> **注意**
> 25 を超えるメトリックまたは 100 万ポイント以上をエクスポートすることはできません。それらをエクスポートした場合は、エラーメッセージを受け取り、エクスポート画面に戻ります。その場合は、より少ない数のメトリックまたは短い期間を選択します。

5. **[エクスポート]** をクリックします。
6. プロンプトが表示されたら、ファイル拡張子 *.xml* でファイル名を指定します。

   指定したメトリックを含む XML ファイルが作成されます。

## メトリックのリフレッシュ

デフォルトでは、モニターの表示は 1 分ごとに自動的に再表示されます。表示の再表示間隔を変更するには、**[ツール]** » **[ユーザー設定]** をクリックします。この設定は、リソースに対する収集間隔とは異なります。リソースに対する収集間隔は、モニタリング対象のリソースからモニターがメトリックを収集する頻度です。「収集間隔」1263 ページを参照してください。

**♦ 再表示間隔の設定**

1. **[ツール]** » **[ユーザー設定]** をクリックします。
2. 必要に応じて設定を変更します。デフォルトは 1 分です。
3. **[保存]** をクリックします。

**[データの再表示]** をクリックすると、最新のメトリックが収集されて表示されます。

**♦ メトリックのリフレッシュ**

● **[データの再表示]** をクリックします。

[F5] キーを押すと、モニターはブラウザーを再ロードし、その時点までに収集されたメトリックを取得して表示します。

**♦ モニターを再ロードします。**

● [F5] キーを押します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

# メトリックのリスト

### SQL Anywhere リソースのメトリックのリスト

| メトリック名 | 説明 |
|---|---|
| [アクティブな HTTP 要求] | HTTP 要求をアクティブに処理している HTTP 接続数を返します。応答を送信した HTTP 接続は含まれません。HttpNumActiveReq サーバープロパティ 760 ページを参照してください。 |
| [アクティブな HTTPS 要求] | HTTPS 要求をアクティブに処理しているセキュア HTTPS 接続数を返します。応答を送信した HTTPS 接続は含まれません。HttpsNumActiveReq サーバープロパティ 761 ページを参照してください。 |
| [監視サーバーステータス] | 次のいずれかの値を返します。<br>● **接続しました。** 監視サーバーはプライマリサーバーに接続している。<br>● **切断しました。** 監視サーバーはプライマリサーバーに接続していない。<br>● **0** この構成には監視サーバーがない。 |
| [ダーティキャッシュ] | ダーティな (書き込みが必要な) キャッシュページの数を返します。CacheFileDirty サーバープロパティ 755 ページを参照してください。 |
| [固定されたキャッシュ] | 固定キャッシュページ数を返します。CachePinned サーバープロパティ 755 ページを参照してください。 |
| [キャッシュ置換] | キャッシュ内の置換されたページ数を返します。CacheReplacements サーバープロパティ 755 ページを参照してください。 |
| [キャッシュサイズ] | 現在のキャッシュサイズ (キロバイト単位) を返します。CurrentCacheSize サーバープロパティ 758 ページを参照してください。 |

| メトリック名 | 説明 |
| --- | --- |
| [接続数] | データベースとの現在の接続数を返します。「sa_conn_info システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| [CPU 使用率] | SQL Anywhere サーバーによって使用された CPU 時間の率を返します。 |
| [データベース名] | データベースの名前を返します。Name データベースプロパティ 799 ページを参照してください。 |
| [ディスク読み取り] | データがディスクから読み込まれている速度を示します (キロバイト数／秒単位)。この値は DiskRead プロパティを基に計算されます。DiskRead データベースプロパティ 788 ページを参照してください。 |
| [ディスク書き込み] | データがディスクに書き込まれている速度を返します (キロバイト数／秒単位)。この値は DiskWrite プロパティを基に計算されます。DiskWrite データベースプロパティ 789 ページを参照してください。 |
| [ファイルサイズ] | メインデータベースファイルのサイズを返します。「事前定義の DB 領域」120 ページを参照してください。 |
| [空きディスク領域] | ディスクの空き領域のサイズを返します。「sa_disk_free_space システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| [ホスト] | データベースサーバーを実行しているコンピューターの名前を返します。通常は、コンピューターのホスト名です。MachineName サーバープロパティ 763 ページを参照してください。 |
| [HTTP 接続] | データベースサーバーで現在開いている HTTP 接続数を返します。この接続は、要求をアクティブに処理しているか、長期 (keep-alive) 接続のキューで待機中である可能性があります。HttpNumConnections サーバープロパティ 760 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| メトリック名 | 説明 |
| --- | --- |
| **[HTTP セッション]** | データベースサーバー内のアクティブおよび休止状態の HTTP セッション数を返します。HttpNumSessions サーバープロパティ 760 ページを参照してください。 |
| **[HTTPS 接続]** | データベースサーバーで現在開いている HTTPS 接続数を返します。この接続は、要求をアクティブに処理しているか、長期 (keep-alive) 接続のキューで待機中である可能性があります。HttpsNumConnections サーバープロパティ 761 ページを参照してください。 |
| **[最後の確認時刻]** | モニターがリソースのデータを最後に収集した時刻を返します。 |
| **[ライセンスタイプ]** | ライセンスタイプを返します。ネットワークシート (per-seat) または CPU ベースのいずれかです。LicenseType サーバープロパティ 763 ページを参照してください。 |
| **[ライセンスを持つ会社]** | ライセンスされた会社の名前を返します。CompanyName サーバープロパティ 757 ページを参照してください。 |
| **[ライセンスされたシート]** | ライセンスされたシートまたはプロセッサーの数を返します。LicenseCount サーバープロパティ 763 ページを参照してください。 |
| **[ライセンスを持つユーザー]** | ライセンスされたユーザーの名前を返します。LicensedUser サーバープロパティ 763 ページを参照してください。 |
| **[長いクエリ情報]** | 実行時間が長いクエリのスレッショルドを超えたクエリを返します。 |
| **[メインヒープページ]** | グローバルサーバーデータ構造に使用されているページ数を返します。MainHeapPages サーバープロパティ 763 ページを参照してください。 |
| **[最大キャッシュサイズ]** | 許容最大キャッシュサイズを返します。MaxCacheSize サーバープロパティ 763 ページを参照してください。 |

| メトリック名 | 説明 |
| --- | --- |
| [ミラーモード] | データベースのミラーリングが使用されていない場合、**[このデータベースではミラーリングが有効になっていません。]** と表示されます。ミラーリングが有効の場合、SET MIRROR OPTION 文で指定された同期モードが同期の場合は **[同期]**、それ以外の場合は **[非同期]** と表示されます。 |
| [ミラーステータス] | 次のいずれかの値を返します。<br>● **同期しています...** ミラーサーバーが接続されていないか、プライマリサーバーのログページの読み込みが完了していない。同期実行モードが非同期である場合にも、この値が返されます。<br>● **同期しました。** ミラーサーバーが接続され、プライマリサーバー上でコミットされたすべての変更が反映されている。 |
| [オペレーティングシステム] | ソフトウェアが実行されているオペレーティングシステムを返します。Platform サーバープロパティ 769 ページを参照してください。 |
| [オペレーティングシステムのバージョン] | ソフトウェアを実行しているオペレーティングシステムの名前を示します。この名前には、ビルド番号やサービスパックなどの情報も含まれています。PlatformVer サーバープロパティ 769 ページを参照してください。 |
| [パートナーステータス] | 次のいずれかの値を返します。<br>● **接続しました。** ミラーサーバーはプライマリサーバーに接続している。<br>● **切断しました。** ミラーサーバーはプライマリサーバーに接続されていない。 |
| [ピークキャッシュサイズ] | 現在のセッションでキャッシュが到達した最大値 (キロバイト単位) を返します。PeakCacheSize サーバープロパティ 769 ページを参照してください。 |
| [プロセッサーのアーキテクチャー] | プロセッサータイプを表す文字列を返します。ProcessorArchitecture サーバープロパティ 771 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| メトリック名 | 説明 |
| --- | --- |
| [処理されたクエリ] | クエリの処理率 (クエリ／秒) を返します。QueryOptimized データベースプロパティ 803 ページ、QueryReused データベースプロパティ 803 ページ、QueryBypassed データベースプロパティ 801 ページを参照してください。 |
| [シート数] | ネットワークサーバーに接続しているユニークなクライアントネットワークアドレスの数を返します。UniqueClientAddresses サーバープロパティ 778 ページを参照してください。 |
| [サーバーの言語] | ロケール言語を示します。これは、データベースサーバーのユーザーによる使用が想定されている言語です。「ロケール言語」485 ページを参照してください。 |
| [サーバー名] | 現在の接続におけるデータベースサーバーの名前を返します。「ServerName (Server) 接続パラメーター」365 ページを参照してください。 |
| [サーバー時間オフセット] | モニターが実行されているコンピューターの時刻と、モニターデータの表示に使用しているコンピューターの時刻の差異が記録されます。「時刻の表示方法の概要」1247 ページを参照してください。 |
| [サーバータイプ] | モニタリングされるデータベースサーバーのタイプ (パーソナルまたはネットワーク) を返します。 |
| [サーバーのバージョン] | 実行中のソフトウェアのバージョンを返します。ProductVersion サーバープロパティ 771 ページを参照してください。 |
| [開始時刻] | SQL Anywhere データベースが起動された時刻を返します。 |
| [総ディスク領域] | ディスクのサイズを返します。 |
| [次の時刻から使用不可] | リソースが使用不可になった時間を返します。 |
| [スケジュールされていない要求] | スケジュールされていない要求の数を返します。UnschReq サーバープロパティ 778 ページを参照してください。 |

| メトリック名 | 説明 |
| --- | --- |
| [未送信のエラーレポート] | 該当データベースについて未送信のエラーレポートの数を返します。エラーレポートは、SQL Anywhere ソフトウェアがクラッシュしたときに送信されます。「リソースからの未送信のエラーレポートに関する警告の抑制」1304 ページを参照してください。 |

### Mobile Link サーバーリソースのメトリックのリスト

| メトリック名 | 説明 |
| --- | --- |
| [同期を認証中] | 現在サーバーで、ユーザー認証フェーズにある要求の数を示します。同期フェーズの詳細については、「-v mlsrv12 オプション」『Mobile Link サーバー管理』を参照してください。 |
| [キャッシュされているリモートタスク要求] | キャッシュされたリモートタスク要求数を示します。「SIRT_NUM_REQUESTS」『Mobile Link サーバー管理』を参照してください。 |
| [コマンドプロセッサーステージ長] | 同期タスクのキューの長さを示します。「CMD_PROCESSOR_STAGE_LEN」『Mobile Link サーバー管理』を参照してください。 |
| [コミット率] | コミットが発生する頻度を示します。 |
| [コミット] | コミットの合計数を示します。「NUM_COMMITS」『Mobile Link サーバー管理』を参照してください。 |
| [同期の完了率] | サーバーの同期の正常完了率を示します。 |
| [接続しているクライアント] | 接続しているクライアント数を示します。「ML_NUM_CONNECTED_CLIENTS」『Mobile Link サーバー管理』を参照してください。 |
| [接続が閉じた割合] | TCP 接続が閉じた割合を示します。 |
| [接続が開いた割合] | TCP 接続が開いた割合を示します。 |
| [接続が拒否された割合] | TCP 接続がサーバーによって拒否された割合を示します。 |
| [統合データベースタイプ] | 統合データベースのタイプを示します。たとえば **SQL Anywhere** のように指定します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| メトリック名 | 説明 |
| --- | --- |
| [統合データベースのバージョン] | 統合データベースのバージョンが表示されます。 |
| [CPU 合計時間] | Mobile Link サーバーによって使用される CPU 時間の量 (マイクロ秒) を示します。「CPU_USAGE」『Mobile Link サーバー管理』を参照してください。 |
| [CPU 使用率] | Mobile Link サーバーによって使用された CPU 時間の割合を示します。 |
| [CPU] | 使用中の CPU の数を示します。 |
| [現在の TCP 接続] | この Mobile Link サーバーで現在使用中の TCP 接続の数を示します。 |
| [使用中のデータベース接続] | 使用中のデータベース接続の数を示します。「DB_CONNECTIONS」『Mobile Link サーバー管理』を参照してください。 |
| [データベースワーカースレッドの数] | データベースワーカースレッドの数を示します。「-w mlsrv12 オプション」『Mobile Link サーバー管理』を参照してください。 |
| [同期をダウンロード中] | ダウンロードの準備、ダウンロードのフェッチ、ダウンロード確認の待機、同期の終了の各フェーズにある現在の同期数を示します。同期フェーズの詳細については、「-v mlsrv12 オプション」『Mobile Link サーバー管理』を参照してください。 |
| [ドライバー名] | 統合データベースのドライバーの名前を示します。 |
| [ドライバーのバージョン] | 統合データベースのドライバーのバージョンを示します。 |
| [エラー] | エラーの合計数を示します。「NUM_ERRORS」『Mobile Link サーバー管理』を参照してください。 |
| [同期の失敗率] | サーバーの同期の失敗率を同期数／秒の単位で示します。 |

| メトリック名 | 説明 |
| --- | --- |
| [失敗した同期] | 失敗した同期の合計数を示します。「NUM_FAILED_SYNCS」『Mobile Link サーバー管理』を参照してください。 |
| [ファイル転送] | 現在接続されている mlfiletransfer 数を示します。「NUM_CONNECTED_FILE_XFERS」『Mobile Link サーバー管理』を参照してください。 |
| [終了した同期の割合] | 完了した同期の割合を示します。 |
| [Mobile Link キャッシュ用ディスク空き領域] | Mobile Link キャッシュ用の、テンポラリディスク上の空きディスク領域をバイト単位で示します。「FREE_DISK_SPACE」『Mobile Link サーバー管理』を参照してください。 |
| [ハートビートステージ長] | 同期以外の定期的なタスクのキューの長さを示します。「HEARTBEAT_STAGE_LEN」『Mobile Link サーバー管理』を参照してください。 |
| ホスト | Mobile Link サーバーを実行しているコンピューターの名前を示します。通常は、コンピューターのホスト名です。 |
| [統合を使用できますか ?] | Mobile Link サーバーが統合データベースに接続できない場合に **[True]** を示します。 |
| [統合データベースはクラスター化されているか ?] | Mobile Link サーバーでクラスター化された統合データベースが使用されている場合は、**True** が表示されます。 |
| [プライマリサーバーはわかっているか ?] | サーバーでプライマリサーバーが認識されている場合は、**[はい]** と表示されます。サーバーでプライマリサーバーが認識されていないか、プライマリサーバーがどれであってもかまわない場合は、**[いいえ]** が表示されます。「PRIMARY_IS_KNOWN」『Mobile Link サーバー管理』を参照してください。 |
| [プライマリサーバーであるか ?] | そのサーバーがプライマリサーバーかである場合は、**[はい]** が表示されます。「SERVER_IS_PRIMARY」『Mobile Link サーバー管理』を参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| メトリック名 | 説明 |
| --- | --- |
| [最後の確認時刻] | モニターがリソースのデータを最後に収集した時刻を示します。 |
| [ライセンスを持つ会社] | ライセンスされた会社の名前を示します。 |
| [ライセンスを持つユーザー] | ライセンスされたユーザーの名前を示します。 |
| [Listener] | 現在接続されている Listener 数を示します。「NUM_CONNECTED_LISTENERS」『Mobile Link サーバー管理』を参照してください。 |
| [最大キャッシュサイズ] | Mobile Link サーバーで使用される最大キャッシュサイズを示します。 |
| [最大クライアント数] | クライアントの最大数を示します。「-sm mlsrv12 オプション」『Mobile Link サーバー管理』を参照してください。 |
| [データベース接続の最大数] | mlsrv12 の -cn オプションまたは -w オプションで設定されたデータベース接続の最大数を示します。「-cn mlsrv12 オプション」『Mobile Link サーバー管理』と「-w mlsrv12 オプション」『Mobile Link サーバー管理』を参照してください。 |
| [最大同期時間] | 同期が発生するまでにかかった最大時間を示します。最も古いアクティブな同期の経過時間 (マイクロ秒)。「LONGEST_SYNC」『Mobile Link サーバー管理』を参照してください。 |
| [最大 TCP 接続数] | mlsrv12 の -nc オプションで設定された TCP 接続の最大数を示します。「-nc mlsrv12 オプション」『Mobile Link サーバー管理』を参照してください。 |
| [許可されるアップロードの最大数] | データベースへの同時アップロードの最大数を示します。「-wu mlsrv12 オプション」『Mobile Link サーバー管理』を参照してください。 |
| [接続の最大待ち時間] | アクティブな同期でデータベースを待機していた最長時間を示します。「LONGEST_DB_WAIT」『Mobile Link サーバー管理』を参照してください。 |

| メトリック名 | 説明 |
| --- | --- |
| [使用メモリ] | 使用中の RAM のバイト数を示します。サーバーが Windows 以外のプラットフォームで実行されている場合は、追跡されたメモリの値を示します。「MEMORY_USED」『Mobile Link サーバー管理』と「TRACKED_MEMORY」『Mobile Link サーバー管理』を参照してください。 |
| [モニター] | 現在接続されている SQL Anywhere モニターおよび Mobile Link モニターの数を示します。「NUM_CONNECTED_MONITORS」『Mobile Link サーバー管理』を参照してください。 |
| [Notifier ステージ長] | Notifier ワークキューの長さを示します。「NOTIFIER_STAGE_LEN」『Mobile Link サーバー管理』を参照してください。 |
| [オペレーティングシステム] | ソフトウェアを実行しているオペレーティングシステムを示します。 |
| [Outbound Enabler ステージ長] | 統合化 Outbound Enabler ワークキューの長さを示します。「OE_STAGE_LEN」『Mobile Link サーバー管理』を参照してください。 |
| [ストリームスタック内のページ] | ネットワークストリームで保持されているページの数を示します。「PAGES_IN_STREAMSTACK」『Mobile Link サーバー管理』を参照してください。 |
| [ロックされているページ] | メモリにロードされているキャッシュページ数を示します。「PAGES_LOCKED」『Mobile Link サーバー管理』を参照してください。 |
| [スワップインしたページ] | ディスクから読み込まれたページの合計数を示します。「PAGES_SWAPPED_IN」『Mobile Link サーバー管理』を参照してください。 |
| [ページのスワップイン率] | ページがディスクから読み込まれる速度を示します。「PAGES_SWAPPED_IN」『Mobile Link サーバー管理』を参照してください。 |
| [スワップアウトしたページ] | ディスクへスワップされたページの合計数を示します。「PAGES_SWAPPED_OUT」『Mobile Link サーバー管理』を参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| メトリック名 | 説明 |
| --- | --- |
| [ページのスワップアウト率] | ページがディスクへスワップされる頻度を示します。「PAGES_SWAPPED_OUT」『Mobile Link サーバー管理』を参照してください。 |
| [使用されているページ] | 使用されたキャッシュページ数。ディスクにスワップされたページが含まれるため、キャッシュサイズよりも大きくなることがあります。「PAGES_USED」『Mobile Link サーバー管理』を参照してください。 |
| [クライアントの接続率] | 現在接続されているクライアントの割合を示します。このメトリックは、**[接続しているクライアント数]** の値を **[クライアントの最大数]** の値で除算し、その結果に 100 を乗算することによって得られます。 |
| [接続の使用率] | 現在使用されている接続の割合を示します。 |
| [ストリームスタック内のページ (%)] | ネットワークストリームで保持されているページの割合を示します。 |
| [ロックされているページ (%)] | ロックされているページの割合を示します。 |
| [使用済みページ (%)] | 使用されているページの割合を示します。ディスクにスワップされたページが含まれるため、キャッシュサイズよりも大きくなることがあります。「PAGES_USED」『Mobile Link サーバー管理』を参照してください。 |
| [ping] | 現在接続されている、ping を実行するクライアント数を示します。「NUM_CONNECTED_PINGS」『Mobile Link サーバー管理』を参照してください。 |
| [プロセッサーのアーキテクチャー] | プロセッサータイプを表す文字列を示します。 |
| [未加工の TCP ステージ長] | ネットワークワークキューの長さを示します。「RAW_TCP_STAGE_LEN」『Mobile Link サーバー管理』を参照してください。 |
| [要求] | 処理されている有効な要求の数を示します。 |

| メトリック名 | 説明 |
| --- | --- |
| [アップロード適用フェーズの要求] | 現在、アップロードの適用フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [ユーザー認証フェーズの要求] | 現在、ユーザー認証フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [同期開始フェーズの要求] | 現在、同期開始フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [ダウンロード確認の接続フェーズの要求] | 現在、ダウンロード確認の接続フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [接続フェーズの要求] | 現在、接続フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [同期終了フェーズの要求] | 現在、同期終了フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [ダウンロードフェッチフェーズの要求] | 現在、ダウンロードのフェッチフェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [ダウンロード確認の DB ワーカーの取得フェーズの要求] | 現在、確認の DB ワーカーの取得フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [DB ワーカーの取得フェーズの要求] | 現在、DB ワーカーの取得フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [非ブロッキングダウンロード確認フェーズの要求] | 現在、非ブロッキングダウンロード確認フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| メトリック名 | 説明 |
| --- | --- |
| [ダウンロード準備フェーズの要求] | 現在、ダウンロードの準備フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [アップロードの受信フェーズの要求] | 現在、アップロードの受信フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [ダウンロード送信フェーズの要求] | 現在、ダウンロードの送信フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [同期要求フェーズの要求] | 現在、同期要求フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [ダウンロード確認の待機フェーズの要求] | 現在、ダウンロード確認の待機フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [ロールバック率] | ロールバックが発生する頻度を示します。「NUM_ROLLBACKS」『Mobile Link サーバー管理』を参照してください。 |
| [ロールバック] | ロールバックの合計数を示します。「NUM_ROLLBACKS」『Mobile Link サーバー管理』を参照してください。 |
| [ローがダウンロードされました] | リモートに送信されたローの合計数を示します。「NUM_ROWS_DOWNLOADED」『Mobile Link サーバー管理』を参照してください。 |
| [ローのダウンロード率] | ローがリモートに送信される頻度を示します。「NUM_ROWS_DOWNLOADED」『Mobile Link サーバー管理』を参照してください。 |
| [ローが更新されました] | リモートから受信したローの合計数を示します。「NUM_ROWS_UPLOADED」『Mobile Link サーバー管理』を参照してください。 |

| メトリック名 | 説明 |
| --- | --- |
| [ローのアップロード率] | リモートからローを受信する頻度を示します。「NUM_ROWS_UPLOADED」『Mobile Link サーバー管理』を参照してください。 |
| [RTNotifier ライトウェイトポーリングヒット率] | リモートタスク要求を最低 1 つ検出したリモートタスクエージェントのライトウェイトポーリング数の率を示します。 |
| [RTNotifier ライトウェイトポーリングのヒット] | リモートタスク要求を最低 1 つ検出したリモートタスクエージェントのライトウェイトポーリングの数を示します。「SIRT_NUM_LWP_HITS」『Mobile Link サーバー管理』を参照してください。 |
| [RTNotifier ライトウェイトポーリングミス率] | リモートタスク要求を検出しなかったリモートタスクエージェントのライトウェイトポーリング数の率を示します。 |
| [RTNotifier ライトウェイトポーリングのミス] | リモートタスクを検出しなかったリモートタスクエージェントのライトウェイトポーリングの数を示します。「SIRT_NUM_LWPS」『Mobile Link サーバー管理』を参照してください。 |
| [RTNotifier ライトウェイトポーリング率] | リモートタスクエージェントのライトウェイトポーリング数の率を示します。 |
| [RTNotifier ライトウェイトポーリング] | リモートタスクエージェントのライトウェイトポーリング数を示します。「SIRT_NUM_REQUESTS」『Mobile Link サーバー管理』を参照してください。 |
| [サーバーキャッシュサイズ] | サーバーキャッシュのサイズを示します。 |
| サーバー名 | 接続されているサーバーの -zs オプションで指定された Mobile Link サーバーの名前を示します。デフォルト値は **<default>** です。「-zs mlsrv12 オプション」『Mobile Link サーバー管理』を参照してください。 |
| [サーバー時間オフセット] | モニターが実行されているコンピューターの時刻と、モニターデータの表示に使用しているコンピューターの時刻の差異を示します。「時刻の表示方法の概要」1247 ページを参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| メトリック名 | 説明 |
| --- | --- |
| [サーバーによって追跡されているメモリ使用率] | サーバーによって割り付けられたメモリの量を示します。このメトリックは、MEMORY_USED メトリックを使用できない Windows 以外のシステムに対して使用します。Microsoft Windows システムでは、精度を上げるために MEMORY_USED メトリックを使用してください。「TRACKED_MEMORY」『Mobile Link サーバー管理』を参照してください。 |
| [開始時刻] | Mobile Link サーバーが起動された時刻を示します。 |
| [同期の開始率] | 同期が開始される頻度を示します。 |
| [ストリームステージ長] | 高レベルのネットワーク処理キューの長さを示します。「STREAM_STAGE_LEN」『Mobile Link サーバー管理』を参照してください。 |
| [成功した同期] | 成功した同期の合計数を示します。「NUM_SUCCESS_SYNCS」『Mobile Link サーバー管理』を参照してください。 |
| [同期エラー率] | 同期エラーの割合を示します。 |
| [同期警告率] | サーバーに対する同期警告の割合を示します。 |
| [同期] | 現在接続されているデータ同期数を示します。NUM_CONNECTED_SYNCS を参照してください。 |
| [同期がブロックされました] | データベース接続を現在待機している同期の数を示します。 |
| [同期が終了しました] | 終了した同期の合計数を示します。 |
| [同期を開始しました] | 開始された同期の合計数を示します。 |
| [読み取られた TCP バイト] | これまでに読み込まれたバイトの合計数を示します。「TCP_BYTES_READ」『Mobile Link サーバー管理』を参照してください。 |
| [TCP バイト読み取り率] | ネットワークからバイトが読み込まれる速度を示します。「TCP_BYTES_READ」『Mobile Link サーバー管理』を参照してください。 |

| メトリック名 | 説明 |
| --- | --- |
| [書き込まれた TCP バイト] | これまでに書き込まれたバイトの合計数を示します。「TCP_BYTES_WRITTEN」『Mobile Link サーバー管理』を参照してください。 |
| [TCP バイト書き込み率] | ネットワークにバイトが書き込まれる速度を示します。「TCP_BYTES_WRITTEN」『Mobile Link サーバー管理』を参照してください。 |
| [TCP 接続が閉じられました] | これまでに閉じられた接続の合計数を示します。「TCP_CONNECTIONS_CLOSED」『Mobile Link サーバー管理』を参照してください。 |
| [TCP 接続が開かれました] | これまでに開かれた接続の合計数を示します。「TCP_CONNECTIONS_OPENED」『Mobile Link サーバー管理』を参照してください。 |
| [TCP 接続が拒否されました] | これまでに拒否された接続の合計数を示します。「TCP_CONNECTIONS_REJECTED」『Mobile Link サーバー管理』を参照してください。 |
| [時間が指定された作業のステージ長] | 動的キャッシュワークキューの長さを示します。「TIMED_WORK_STAGE_LEN」『Mobile Link サーバー管理』を参照してください。 |
| [次の時刻から使用不可] | リソースが使用不可になった時間を示します。 |
| [不明な接続済みクライアント] | 接続しているクライアントのうち、オリジンが不明なクライアントの数を示します。 |
| [未送信のエラーレポート] | 該当サーバーについて未送信のエラーレポートの数を示します。エラーレポートは、SQL Anywhere 12 ソフトウェアがクラッシュしたときに送信されます。「リソースからの未送信のエラーレポートに関する警告の抑制」1304 ページを参照してください。 |
| [使用中のアップロード接続] | 現在使用中のアップロード接続の数を示します。「NUM_UPLOAD_CONNS_IN_USE」『Mobile Link サーバー管理』を参照してください。 |
| [同期を更新中] | アップロードフェーズにある同期の数を示します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| メトリック名 | 説明 |
| --- | --- |
| [バージョン] | 実行中のソフトウェアのバージョンを示します。 |
| [VM メモリ使用率] | 付加された VM で使用されるメモリの量を示します。「VM_MEM_USE」『Mobile Link サーバー管理』を参照してください。 |
| [データベース接続の待機中] | データベース接続を待機している要求数。 |
| [警告] | 警告の合計数を示します。「NUM_WARNINGS」『Mobile Link サーバー管理』を参照してください。 |

### Mobile Link サーバーファームリソースのメトリックのリスト

| メトリック名 | スコープ | 説明 |
| --- | --- | --- |
| [同期を認証中] | Average | 現在サーバーで、ユーザー認証フェーズにある要求の数を示します。同期フェーズの詳細については、「-v mlsrv12 オプション」『Mobile Link サーバー管理』を参照してください。 |
| [バイト読み取り率] | Average | ネットワークからバイトが読み込まれる速度を示します。「TCP_BYTES_READ」『Mobile Link サーバー管理』を参照してください。 |
| [バイト書き込み率] | Average | ネットワークにバイトが書き込まれる速度を示します。「TCP_BYTES_WRITTEN」『Mobile Link サーバー管理』を参照してください。 |
| [キャッシュされているリモートタスク要求] | Average | キャッシュされたリモートタスク要求数を示します。「SIRT_NUM_REQUESTS」『Mobile Link サーバー管理』を参照してください。 |

| メトリック名 | スコープ | 説明 |
| --- | --- | --- |
| [コマンドプロセッサーステージ長] | Average | 同期タスクのキューの長さを示します。「CMD_PROCESSOR_STAGE_LEN」『Mobile Link サーバー管理』を参照してください。 |
| [コミット率] | Average | コミットが発生する頻度を示します。 |
| [コミット] | Average | コミットの合計数を示します。「NUM_COMMITS」『Mobile Link サーバー管理』を参照してください。 |
| [同期の完了率] | Total | サーバーの同期の正常完了率を示します。 |
| [接続しているクライアント] | Average | 接続しているクライアント数を示します。「ML_NUM_CONNECTED_CLIENTS」『Mobile Link サーバー管理』を参照してください。 |
| [接続が閉じた割合] | Average | TCP 接続が閉じた割合を示します。 |
| [接続が開いた割合] | Average | TCP 接続が開いた割合を示します。 |
| [接続が拒否された割合] | Average | 接続がサーバーによって拒否された割合を示します。 |
| [CPU 合計時間] | Total | Mobile Link サーバーによって使用される CPU 時間の量 (マイクロ秒) を示します。「CPU_USAGE」『Mobile Link サーバー管理』を参照してください。 |
| CPU 使用率 | Average | Mobile Link サーバーによって使用された CPU の平均量を示します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| メトリック名 | スコープ | 説明 |
|---|---|---|
| [現在の TCP 接続] | Total | この Mobile Link サーバーで現在使用中の TCP 接続の数を示します。 |
| [使用中のデータベース接続] | Total | 使用中のデータベース接続の数を示します。「DB_CONNECTIONS」『Mobile Link サーバー管理』を参照してください。 |
| [同期をダウンロード中] | Average | ダウンロードの準備、ダウンロードのフェッチ、ダウンロード確認の待機、同期の終了の各フェーズにある現在の同期数を示します。同期フェーズの詳細については、「-v mlsrv12 オプション」『Mobile Link サーバー管理』を参照してください。 |
| [エラー] | Average | エラーの平均数を示します。「NUM_ERRORS」『Mobile Link サーバー管理』を参照してください。 |
| [同期の失敗率] | Total | サーバーの失敗した同期の割合を示します。 |
| [失敗した同期] | Average | 失敗した同期の合計数を示します。「NUM_FAILED_SYNCS」『Mobile Link サーバー管理』を参照してください。 |
| [ファイル転送] | Total | 現在接続されている mlfiletransfer 数を示します。「NUM_CONNECTED_FILE_XFERS」『Mobile Link サーバー管理』を参照してください。 |
| [終了した同期の割合] | Total | 完了した同期の割合を示します。 |

| メトリック名 | スコープ | 説明 |
| --- | --- | --- |
| **[Mobile Link キャッシュ用ディスク空き領域]** | Average | Mobile Link キャッシュ用の、テンポラリディスク上の空きディスク領域をバイト単位で示します。「FREE_DISK_SPACE」『Mobile Link サーバー管理』を参照してください。 |
| **[ハートビートステージ長]** | Average | 同期以外の定期的なタスクのキューの長さを示します。「HEARTBEAT_STAGE_LEN」『Mobile Link サーバー管理』を参照してください。 |
| **[Listener]** | Total | 現在接続されている Listener 数を示します。「NUM_CONNECTED_LISTENERS」『Mobile Link サーバー管理』を参照してください。 |
| **[最大同期時間]** | Maximum | 同期が発生するまでにかかった最大時間を示します。最も古いアクティブな同期の経過時間 (マイクロ秒)。「LONGEST_SYNC」『Mobile Link サーバー管理』を参照してください。 |
| **[接続の最大待ち時間]** | Maximum | アクティブな同期でデータベースを待機していた最長時間を示します。「LONGEST_DB_WAIT」『Mobile Link サーバー管理』を参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| メトリック名 | スコープ | 説明 |
| --- | --- | --- |
| [使用メモリ] | Average | 使用中の RAM のバイト数を示します。サーバーが Windows 以外のプラットフォームで実行されている場合は、追跡されたメモリの値を示します。「MEMORY_USED」『Mobile Link サーバー管理』と「TRACKED_MEMORY」『Mobile Link サーバー管理』を参照してください。 |
| [モニター] | Total | 現在接続されている SQL Anywhere モニターおよび Mobile Link モニターの数を示します。「NUM_CONNECTED_MONITORS」『Mobile Link サーバー管理』を参照してください。 |
| [Notifier ステージ長] | Average | Notifier ワークキューの長さを示します。「NOTIFIER_STAGE_LEN」『Mobile Link サーバー管理』を参照してください。 |
| [Outbound Enabler ステージ長] | Average | 統合化 Outbound Enabler ワークキューの長さを示します。「OE_STAGE_LEN」『Mobile Link サーバー管理』を参照してください。 |
| [ストリームスタック内のページ] | Average | ネットワークストリームで保持されているページの数を示します。「PAGES_IN_STREAMSTACK」『Mobile Link サーバー管理』を参照してください。 |
| [ロックされているページ] | Average | メモリにロードされているキャッシュページ数を示します。「PAGES_LOCKED」『Mobile Link サーバー管理』を参照してください。 |

| メトリック名 | スコープ | 説明 |
| --- | --- | --- |
| [スワップインしたページ] | Average | ディスクから読み込まれたページの合計数を示します。「PAGES_SWAPPED_IN」『Mobile Link サーバー管理』を参照してください。 |
| [ページのスワップイン率] | Average | ページがディスクから読み込まれる速度を示します。「PAGES_SWAPPED_IN」『Mobile Link サーバー管理』を参照してください。 |
| [スワップアウトしたページ] | Average | ディスクへスワップされたページの合計数を示します。「PAGES_SWAPPED_OUT」『Mobile Link サーバー管理』を参照してください。 |
| [ページのスワップアウト率] | Average | ページがディスクへスワップされる頻度を示します。「PAGES_SWAPPED_OUT」『Mobile Link サーバー管理』を参照してください。 |
| [使用されているページ] | Average | 使用されたキャッシュページ数。ディスクにスワップされたページが含まれるため、キャッシュサイズよりも大きくなることがあります。「PAGES_USED」『Mobile Link サーバー管理』を参照してください。 |
| [クライアントの接続率] | Average | 現在接続されているクライアントの割合を示します。このメトリックは、**[接続しているクライアント数]** の値を **[クライアントの最大数]** の値で除算し、その結果に 100 を乗算することによって得られます。 |
| [接続の使用率] | Average | 現在使用されている接続の割合を示します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| メトリック名 | スコープ | 説明 |
| --- | --- | --- |
| **[ストリームスタック内のページ (%)]** | Average | ネットワークストリームで保持されているページの割合を示します。 |
| **[ロックされているページ (%)]** | Average | ロックされているページの割合を示します。 |
| **[使用済みページ (%)]** | Average | 使用されているページの割合を示します。ディスクにスワップされたページが含まれるため、キャッシュサイズよりも大きくなることがあります。PAGES_USED を参照してください。 |
| **[ping]** | Total | 現在接続されている、ping を実行するクライアント数を示します。「NUM_CONNECTED_PINGS」『Mobile Link サーバー管理』を参照してください。 |
| **[未加工の TCP ステージ長]** | Average | ネットワークワークキューの長さを示します。「RAW_TCP_STAGE_LEN」『Mobile Link サーバー管理』を参照してください。 |
| **[要求]** | Total | 処理されている有効な要求の数を示します。 |
| **[アップロード適用フェーズの要求]** | Average | 現在、アップロードの適用フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| **[ユーザー認証フェーズの要求]** | Average | 現在、ユーザー認証フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |

| メトリック名 | スコープ | 説明 |
| --- | --- | --- |
| [同期開始フェーズの要求] | Average | 現在、同期開始フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [ダウンロード確認の接続フェーズの要求] | Average | 現在、ダウンロード確認の接続フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [接続フェーズの要求] | Average | 現在、接続フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [同期終了フェーズの要求] | Average | 現在、同期終了フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [ダウンロードフェッチフェーズの要求] | Average | 現在、ダウンロードのフェッチフェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [ダウンロード確認の DB ワーカーの取得フェーズの要求] | Average | 現在、確認の DB ワーカーの取得フェーズにある同期の数を示します。「NUM_IN_GET_DB_WORKER_FOR_ACK」『Mobile Link サーバー管理』を参照してください。 |
| [DB ワーカーの取得フェーズの要求] | Average | 現在、DB ワーカーの取得フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| メトリック名 | スコープ | 説明 |
| --- | --- | --- |
| [非ブロッキングダウンロード確認フェーズの要求] | Average | 現在、非ブロッキングダウンロード確認フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [ダウンロード準備フェーズの要求] | Average | 現在、ダウンロードの準備フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [アップロードの受信フェーズの要求] | Average | 現在、アップロードの受信フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [ダウンロード送信フェーズの要求] | Average | 現在、ダウンロードの送信フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [同期要求フェーズの要求] | Average | 現在、同期要求フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [ダウンロード確認の待機フェーズの要求] | Average | 現在、ダウンロード確認の待機フェーズにある同期の数を示します。mlsrv12 -vm オプション『Mobile Link サーバー管理』を参照してください。 |
| [ロールバック率] | Average | ロールバックの合計数を示します。「NUM_ROLLBACKS」『Mobile Link サーバー管理』を参照してください。 |
| [ロールバック] | Average | ロールバックの合計数を示します。「NUM_ROLLBACKS」『Mobile Link サーバー管理』を参照してください。 |

| メトリック名 | スコープ | 説明 |
| --- | --- | --- |
| [ローがダウンロードされました] | Average | リモートに送信されたローの合計数を示します。「NUM_ROWS_DOWNLOADED」『Mobile Link サーバー管理』を参照してください。 |
| [ローのダウンロード率] | Average | リモートに送信されたローの合計数を示します。「NUM_ROWS_DOWNLOADED」『Mobile Link サーバー管理』を参照してください。 |
| [ローが更新されました] | Average | リモートから受信したローの合計数を示します。「NUM_ROWS_UPLOADED」『Mobile Link サーバー管理』を参照してください。 |
| [ローのアップロード率] | Average | リモートから受信したローの合計数を示します。「NUM_ROWS_UPLOADED」『Mobile Link サーバー管理』を参照してください。 |
| [RTNotifier ライトウェイトポーリングヒット率] | Average | リモートタスク要求を最低 1 つ検出したリモートタスクエージェントのライトウェイトポーリング数の率を示します。 |
| [RTNotifier ライトウェイトポーリングのヒット] | Average | リモートタスク要求を最低 1 つ検出したリモートタスクエージェントのライトウェイトポーリングの数を示します。「SIRT_NUM_LWP_HITS」『Mobile Link サーバー管理』を参照してください。 |
| [RTNotifier ライトウェイトポーリングミス率] | Average | リモートタスク要求を検出しなかったリモートタスクエージェントのライトウェイトポーリング数の率を示します。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| メトリック名 | スコープ | 説明 |
| --- | --- | --- |
| **[RTNotifier ライトウェイトポーリングのミス]** | Average | リモートタスクを検出しなかったリモートタスクエージェントのライトウェイトポーリングの数を示します。「SIRT_NUM_LWPS」『Mobile Link サーバー管理』を参照してください。 |
| **[RTNotifier ライトウェイトポーリング率]** | Average | リモートタスクエージェントのライトウェイトポーリング数の率を示します。 |
| **[RTNotifier ライトウェイトポーリング]** | Average | リモートタスクエージェントのライトウェイトポーリング数を示します。「SIRT_NUM_REQUESTS」『Mobile Link サーバー管理』を参照してください。 |
| **[サーバーキャッシュサイズ]** | Average | サーバーキャッシュのサイズを示します。 |
| **[サーバーによって追跡されているメモリ使用率]** | Average | サーバーによって割り付けられたメモリの量を示します。このメトリックは、MEMORY_USED メトリックを使用できない Windows 以外のシステムに対して使用します。Microsoft Windows システムでは、精度を上げるために MEMORY_USED メトリックを使用してください。「TRACKED_MEMORY」『Mobile Link サーバー管理』を参照してください。 |
| **[同期の開始率]** | Total | 同期が開始される頻度を示します。 |
| **[ストリームステージ長]** | Average | 高レベルのネットワーク処理キューの長さを示します。「STREAM_STAGE_LEN」『Mobile Link サーバー管理』を参照してください。 |

| メトリック名 | スコープ | 説明 |
| --- | --- | --- |
| [成功した同期] | Average | 成功した同期の合計数を示します。「NUM_SUCCESS_SYNCS」『Mobile Link サーバー管理』を参照してください。 |
| [同期エラー率] | Total | 同期エラーの割合を示します。 |
| [同期警告率] | Total | サーバーに対する同期警告の割合を示します。 |
| [同期] | Total | 現在接続されているデータ同期数を示します。「NUM_CONNECTED_SYNCS」『Mobile Link サーバー管理』を参照してください。 |
| [同期がブロックされました] | Average | データベース接続を現在待機している同期の数を示します。 |
| [同期が終了しました] | Total | 完了した同期の数が表示されます。 |
| [同期を開始しました] | Total | 開始された同期の合計数を示します。 |
| [読み取られた TCP バイト] | Average | これまでに読み込まれたバイトの合計数を示します。「TCP_BYTES_READ」『Mobile Link サーバー管理』を参照してください。 |
| [書き込まれた TCP バイト] | Average | これまでに書き込まれたバイトの合計数を示します。「TCP_BYTES_WRITTEN」『Mobile Link サーバー管理』を参照してください。 |
| [TCP 接続が閉じられました] | Average | これまでに閉じられた接続の合計数を示します。「TCP_CONNECTIONS_CLOSED」『Mobile Link サーバー管理』を参照してください。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| メトリック名 | スコープ | 説明 |
|---|---|---|
| [TCP 接続が開かれました] | Average | これまでに開かれた接続の合計数を示します。「TCP_CONNECTIONS_OPENED」『Mobile Link サーバー管理』を参照してください。 |
| [TCP 接続が拒否されました] | Average | これまでに拒否された接続の合計数を示します。「TCP_CONNECTIONS_REJECTED」『Mobile Link サーバー管理』を参照してください。 |
| [時間が指定された作業のステージ長] | Average | 動的キャッシュワークキューの長さを示します。「TIMED_WORK_STAGE_LEN」『Mobile Link サーバー管理』を参照してください。 |
| [不明な接続済みクライアント] | Average | 接続しているクライアントのうち、オリジンが不明なクライアントの数を示します。 |
| [使用中のアップロード接続] | Average | 現在使用中のアップロード接続の数を示します。「NUM_UPLOAD_CONNS_IN_USE」『Mobile Link サーバー管理』を参照してください。 |
| [同期を更新中] | Average | アップロードフェーズにある同期の数を示します。 |
| [VM メモリ使用率] | Average | 付加された VM で使用されるメモリの量を示します。「VM_MEM_USE」『Mobile Link サーバー管理』を参照してください。 |
| [データベース接続の待機中] | Average | データベース接続を待機している要求数。 |
| [警告] | Average | 警告の平均数を示します。「NUM_WARNINGS」『Mobile Link サーバー管理』を参照してください。 |

### Relay Server ファームリソースのメトリックのリスト

| メトリック名 | 説明 |
|---|---|
| アクティブな要求 | 同時要求の合計数を示します。 |
| バックエンドサーバーの平均処理時間 | バックエンドサーバーが HTTP 要求の処理に要した平均時間を示します。 |
| 平均要求完了時間 | クライアントに応答が送信された後、バックエンド接続が閉じるまで Relay Server が待機する平均時間を示します。 |
| 平均要求時間 | クライアントから HTTP 要求全体を受信するまで Relay Server が待機する平均時間を示します。 |
| 要求と応答の平均サイクル時間 | Relay Server が要求を処理し、応答を受信するのに要する平均時間を示します。 |
| 平均応答時間 | Relay Server がバックエンドサーバーから HTTP 応答全体を受信するのに要する平均時間を示します。 |
| チャネルタイムアウト | Relay Server がバックエンドサービスのブラックアウトを検出するまでの最大遅延時間。 |
| CPU 使用率 | Relay Server によって使用された CPU 時間の平均割合を示します。 |
| 下りチャネルのバイト | 前回の収集期間に、Outbound Enabler からバックエンドサーバーの拡張機能へ送信されたバイトの合計数を示します。 |
| エラー数 | 前回の収集期間に発生したエラーの合計数を示します。この数には、子のエラー数が含まれます。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| メトリック名 | 説明 |
| --- | --- |
| ファームのステータス | Relay Server ファームの現在のステータスを示します。<br>● **0** Relay Server ファームは有効であり、正常にサービスを提供できます。<br>● **1** Relay Server ファームは無効であるため、Outbound Enabler は接続できず、サービスを提供できません。クライアントもこのバックエンドファームに Relay Server 経由でアクセスできません。<br>● **2** Relay Server ファームは有効ですが、Outbound Enabler の一部は Relay Servers ファームのすべての Relay Server に接続されているわけではありません。サービスの質に影響する可能性があります。<br>● **3** Relay Server ファームは有効ですが、どのバックエンドサーバーも接続されていないため、クライアントはこのバックエンドファームにアクセスできません。 |
| ログファイル用の空きディスク領域 | Relay Server ファームのログファイルに使用可能なディスク領域を示します。 |
| ホスト | Relay Server ファームを実行しているコンピューターの名前を示します。 |
| オペレーティングシステムのバージョン | Relay Server ソフトウェアを実行しているオペレーティングシステムの名前を示します。この名前には、ビルド番号やサービスパックなどの情報も含まれています。 |
| ピーク時のアクティブな要求 | Relay Server が起動されてからの同時要求の最大数を示します。 |
| ピーク時の共有メモリ使用率 | Relay Server が起動されてからの Relay Server による共有メモリの最大使用率を示します。 |
| プロセッサーのアーキテクチャー | Relay Server ファームが実行されているプロセッサーのタイプを示します。 |

| メトリック名 | 説明 |
| --- | --- |
| Relay Server の開始時刻 | Relay Server の起動時刻を示します。バックエンドサーバーの場合、Outbound Enabler がバックエンドサーバーの拡張機能に接続された時刻を示します。 |
| Relay Server のステータス | Relay Server のステータスを示します。 |
| Relay Server のバージョン | 実行中の Relay Server ソフトウェアのバージョンを示します。 |
| 要求バイト | クライアントからクライアントの拡張機能への HTTP 要求からのバイトの合計数を示します。 |
| 完了した要求 | 終了した要求応答サイクルの合計数を示します。 |
| 応答バイト | クライアントの拡張機能からクライアントへの HTTP 応答のバイトの合計数を示します。 |
| 作成されたセッション | 作成された新しい HTTP セッションの合計数を示します。 |
| 共有メモリの制限 | Relay Server が使用できる共有メモリの合計を示します。 |
| 次の時刻から使用不可 | リソースが使用不可になった時間を示します。 |
| 上りチャネルのバイト | バックエンドサーバーの拡張機能から Outbound Enabler へ送信されたバイトの合計数を示します。 |
| Web サーバーのバージョン | 実行中の Web サーバーソフトウェアのバージョンを示します。 |

# モニターのユーザー

モニターにログインするには、ユーザー名とパスワードを使用します。モニターにログインするためのユーザー ID とパスワードでは、大文字と小文字が区別されます。

モニターは次の 3 種類のユーザーをサポートしています。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| ユーザーのタイプ | 説明 |
| --- | --- |
| 読み込み専用ユーザー | モニターリソースに対する読み込み専用アクセスがあります。読み込み専用ユーザーはメトリックを表示できますが、**[管理]** ウィンドウにはアクセスできません。ログイン画面からユーザーを作成した場合、このユーザーは読み込み専用ユーザーとなります。ユーザー名とパスワードが必要です。 |
| オペレーター | モニターリソースに対する読み込み専用アクセスがあり、警告を受信できます。このユーザーはメトリックの表示、電子メール警告の受信、および警告の解決と削除ができます。ただし、**[管理]** ウィンドウにはアクセスできません。ユーザー名とパスワードが必要です。 |
| 管理者 | オペレーターと同じアクセス権を持ち、さらにリソースの設定とユーザーの追加ができます。また、**[管理]** ウィンドウにもアクセスできます。デフォルトユーザーである **admin** は管理者です。ユーザー名とパスワードが必要です。 |

すべてのユーザーは、ログイン後、各自のユーザーのタイプを確認し、設定を変更できます。ただし、管理者のみがユーザーのタイプを変更できます。「モニターユーザーの編集」1299 ページを参照してください。

**♦ モニターユーザータイプの確認**

1. モニターにログインします。
2. **[ツール]** » **[ユーザー設定]** をクリックします。
3. ウィンドウ内の **[ユーザーのタイプ]** 設定を確認します。

## デフォルトユーザー

モニターの初回起動時には、デフォルトでユーザー名が **admin** でパスワードが **admin** の管理者ユーザーが 1 つ用意されています。このユーザーにはデフォルトですべてのパーミッションが与えられています。デフォルトの管理者パスワードを変更して、モニターへのアクセスを制限することをおすすめします。「モニターユーザーの編集」1299 ページを参照してください。

## 参照

● 「[管理] ウィンドウ」1247 ページ

## モニターユーザーの作成

デフォルトでは、どのユーザーもログイン画面から各自の読み込み専用ユーザーを作成し、モニターへ読み込み専用アクセスすることができます。ただし、管理者は、管理者だけがユーザーを作成できるようにこのデフォルトを変更できます。「管理者だけがユーザーを作成できるようにする」1301 ページを参照してください。

**♦ ログイン画面から読み込み専用ユーザーを作成するには、次の手順に従います。**

1. **[新しいユーザーを作成]** をクリックします。

> **注意**
> **[新しいユーザーを作成]** リンクを使用できない場合は、管理者だけが新しいユーザーを作成できるように、管理者によってデフォルトの動作が変更されています。新しいユーザーの作成については、モニターの管理者に作成を依頼してください。

2. 新しいユーザーの情報を入力します。

   電子メールアドレスは、モニターからの電子メール警告を受信する場合にだけ必要です。ユーザーのタイプをオペレーターまたは管理者に変更し、電子メール警告を受信するように登録するには、モニターの管理者に問い合わせてください。「警告の電子メール送信」1307 ページを参照してください。

ログインすると、管理者は、他の管理者を含むすべてのタイプのユーザーを作成できます。

**♦ 管理者としてログインしたときに新しいユーザーを作成するには、次の手順に従います。**

1. モニターに管理者としてログインします。
2. **[ツール]** » **[管理]** をクリックします。
3. **[ユーザー]** をクリックします。
4. **[新規]** をクリックします。
5. 新しいユーザーの情報を入力します。電子メールアドレスは、モニターからの電子メール警告を受信するユーザーにだけ必要です。
6. **[使用する言語タイプ]** ドロップダウンメニューから言語をクリックします。指定した言語は、警告で使用される言語などの、モニターで使用される言語に設定されます。
7. ユーザータイプをクリックします。それぞれのタイプには、異なるパーミッションが付与されています。「モニターのユーザー」1296 ページを参照してください。
8. **[次へ]** をクリックします。
9. このユーザーのリソースダッシュボードをクリックします。
10. **[保存]** をクリックします。
11. **[閉じる]** をクリックします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

12. オペレーターユーザーまたは管理者ユーザーを作成した場合、このユーザーは電子メールで警告を受け取ることができます。「警告の電子メール送信」1307 ページを参照してください。

**参照**

- 「モニターユーザーの編集」1299 ページ
- 「管理者だけがユーザーを作成できるようにする」1301 ページ

## モニターユーザーとリソースの関連付け

**♦ オペレーターまたは管理者をリソースと関連付けるには、次の手順に従います。**

オペレーターユーザーまたは管理者ユーザーがリソースに関する電子メール警告を受信するようにするには、そのユーザーを該当するリソースに関連付ける必要があります。

1. モニターに管理者としてログインします。
2. **[ツール]** » **[管理]** をクリックします。
3. オペレーターまたは管理者のユーザーアカウントに電子メールアドレスが指定されていることを確認します。

   **[ユーザー]** をクリックして、ユーザーに電子メールアドレスが指定されていることを確認します。
4. **[リソース]** をクリックして、リストからリソースを選択し、**[設定]** をクリックします。
5. **[オペレーター]** をクリックします。
6. **[選択可能なオペレーター]** リストでユーザーをクリックし、**[追加]** をクリックします。

   **[選択されたオペレーター]** リストにユーザーの名前が表示されます。
7. **[保存]** をクリックします。
8. **[閉じる]** をクリックします。
9. モニターが警告の通知を電子メールで送信するよう設定されていることを確認します。「モニターによる警告電子メールの送信の有効化」1307 ページを参照してください。

**参照**

- 「警告の電子メール送信」1307 ページ

## モニターユーザーの編集

**♦ 既存のモニターユーザーを編集するには、次の手順に従います。**

すべてのユーザーは、ログイン後、**[ツール]** » **[ユーザー設定]** をクリックして各自のユーザー設定を変更できます。ただし、管理者のみがユーザーのタイプを変更できます。

1. モニターに管理者としてログインします。
2. **[ツール]** » **[管理]** をクリックします。
3. **[ユーザー]** をクリックします。
4. 編集するユーザーをクリックし、**[編集]** をクリックします。
5. 必要に応じてユーザーの設定を変更します。電子メールアドレスは、モニターからの電子メール警告を受信するユーザーにだけ必要です。
6. **[保存]** をクリックします。
7. **[閉じる]** をクリックします。
8. オペレーターユーザーまたは管理者ユーザーを作成した場合、このユーザーは電子メールで警告を受け取ることができます。「警告の電子メール送信」1307 ページを参照してください。

**参照**


- 「モニターユーザーの作成」1298 ページ
- 「モニターユーザーの削除」1300 ページ


## モニターユーザーの削除

♦ **既存のモニターユーザーを削除するには、次の手順に従います。**

ユーザーを削除すると、そのユーザーがモニターから削除され、リソースとの関連付けがすべて破棄されます。

1. モニターに管理者としてログインします。
2. **[ツール]** » **[管理]** をクリックします。
3. **[ユーザー]** をクリックします。
4. ユーザーをクリックして **[削除]** をクリックします。
5. **[はい]** をクリックして、選択したユーザーを削除します。

   ユーザーがモニターから削除されます。
6. **[閉じる]** をクリックします。

**参照**


- 「モニターユーザーの作成」1298 ページ
- 「モニターユーザーの編集」1299 ページ

## 管理者だけがユーザーを作成できるようにする

♦ **ユーザーの作成を管理者に限定するには、次の手順に従います。**

デフォルトでは、どのユーザーもログイン画面から各自のユーザー名とパスワードを作成し、モニターへ読み込み専用アクセスすることができます。ただし、管理者は、管理者だけがユーザーを作成できるように、この機能をオフにします。

1. モニターに管理者としてログインします。
2. **[ツール]** » **[管理]** をクリックします。
3. **[設定]** をクリックします。
4. **[編集]** をクリックします。
5. **[オプション]** をクリックします。
6. **[すべてのユーザーに SQL Anywhere モニターの読み込み専用アクセスを許可します。]** オプションをオフにします。
7. **[保存]** をクリックします。
8. **[閉じる]** をクリックします。

**参照**

- 「モニターユーザーの作成」1298 ページ
- 「モニターユーザーの編集」1299 ページ

# 警告

ここで言う「警告」とは、管理者またはオペレーターによる注目を要するリソースに関する特定の状況または状態のことです。警告は、収集されたメトリックを基にモニターによって検出されます。モニターリングされているリソースでは検出されません。

空きディスク領域の減少、ログイン試行の失敗、高メモリ使用率などの状況については、事前に定義された警告があります。リソースを編集すると、デフォルトのスレッショルド値を変更できます。「警告スレッショルドの指定」1304 ページを参照してください。

警告の条件が満たされると、その警告が、指定したリソースの **[警告リスト]** ウィジェットにリストされます。また、警告が発生したときにオペレーターや管理者に電子メールが送信されるよう SQL Anywhere モニターを設定することもできます。「警告の電子メール送信」1307 ページを参照してください。

デフォルトで、警告は **[警告リスト]** ウィジェットに表示され、警告には、問題の原因に関する情報や、問題解決のアドバイスなどが含まれます。**[リソースリスト]** で、リソースのステータスは警告の存在と重大度を反映して変更されます。

1 リソースにつき最新の 50 個の警告のみがモニターによって格納され、**[警告リスト]** ウィジェットに表示されます。それよりも古い警告は削除されます。

## 警告の表示

♦ **警告を表示するには、次の手順に従います。**

1 リソースにつき最新の 50 個の警告だけがモニターによって格納され、**[警告リスト]** に表示されます。

警告はログインしたあらゆるユーザーが表示できますが、警告の解決と削除ができるのはオペレーターと管理者だけです。

1. **[警告]** » **[今日]** をクリックします。
2. リストで警告をクリックし、**[詳細]** をクリックします。
3. **[OK]** をクリックします。

**参照**

- 「警告の解決」1302 ページ
- 「警告の削除」1303 ページ
- 「警告の電子メール送信」1307 ページ

## 警告の解決

♦ **警告を解決するには、次の手順に従います。**

オペレーターまたは管理者は、警告の原因となった問題に対応したら、その警告を解決済みとマークできます。

警告を解決すると、警告の **[ステータス]** が **[*user-name* によって解決済み]** に変更されますが、警告そのものは警告リストに残ります。警告をリストから削除するには、該当する警告を削除する必要があります。「警告の削除」1303 ページを参照してください。

1. モニターに管理者ユーザーまたはオペレーターユーザーとしてログインします。
2. **[概要]** をクリックします。
3. **[警告リスト]** ウィジェットで、リストで警告をクリックし、**[解決済みとマークする]** をクリックして、選択された警告を解決します。

   **[ステータス]** カラムの値は、**[*your-user-name* によって解決済み]** に変わります。

   その警告が該当リソースの唯一の未解決警告だった場合、**[リソースリスト]** ウィジェットのリソースのステータスは正常 (アイコンなし) に変わります。

**参照**


- 「[概要] ダッシュボード」1240 ページ
- 「警告の削除」1303 ページ
- 「警告の電子メール送信」1307 ページ
- 「警告の表示」1302 ページ
- 「警告」1301 ページ


## 警告の削除

**♦ 警告の削除**

オペレーターまたは管理者は、**[警告リスト]** ウィジェットから任意の警告を削除できます。警告は、そのモニターの状態に関係なく削除できます。**[警告リスト]** ウィジェットから警告を削除しても、警告の電子メールには影響はありません。同様に、警告の電子メールを削除しても、**[警告リスト]** ウィジェットには影響しません。

1. モニターに管理者ユーザーまたはオペレーターユーザーとしてログインします。
2. **[概要]** をクリックします。
3. **[警告リスト]** ウィジェットで、リストから警告をクリックし、**[削除]** をクリックします。
4. **[警告の削除]** ウィンドウが表示され、次のオプションから 1 つを選択するように要求されます。
   - **選択した警告を削除する**　このオプションは、**[警告リスト]** ウィジェットで選択した警告を削除するためだけに選択します。
   - **次より前に受信した警告を削除する**　指定した時間より前に受信したすべての警告を削除するには、このオプションを選択します。このオプションの隣にある日付をクリックして日付を指定すると、それよりも前の警告を削除できます。このオプションは、**[警告リスト]** ウィジェットに含まれているリソースの警告を削除するだけです。
   - **次のリソースの警告を削除する**　選択したリソースのすべての警告を削除するには、このオプションを選択します。

   **[OK]** をクリックします。
5. プロンプトが表示されたら、**[はい]** をクリックして、削除を確認します。

   **[警告リスト]** ウィジェットから警告が削除されます。

**参照**


- 「警告の解決」1302 ページ
- 「警告の電子メール送信」1307 ページ
- 「警告の表示」1302 ページ
- 「警告」1301 ページ

## 警告スレッショルドの指定

♦ **警告を発行するタイミングを設定するには、次の手順に従います。**

管理者は、警告を発行するタイミングを設定できます。ただし、モニターが収集するメトリックを設定したり、デフォルトリソースの SQL Anywhere モニターを設定することはできません。

1. モニターに管理者としてログインします。
2. **[ツール]** » **[管理]** をクリックします。
3. **[リソース]** をクリックし、リストからリソースを選択します。
4. **[設定]** をクリックします。
5. **[警告のスレッショルド]** をクリックします。スレッショルドを編集します。警告の定義については、「警告スレッショルド」1305 ページを参照してください。
6. その他の設定を必要に応じて指定します。
7. **[保存]** をクリックします。
8. 認証情報を要求されたら、データベースの DBA ユーザー ID とパスワードを指定します。

   SQL Anywhere データベースのリソースを編集する場合は、データベースの DBA ユーザー ID とパスワードを指定する必要があります。これらのクレデンシャルは、データベースへの接続と、モニタリングに必要なインストールされたデータベースオブジェクトの変更に使用されます。これにより、DBA クレデンシャルがモニターから削除されます。
9. **[閉じる]** をクリックします。
10. **[閉じる]** をクリックします。

**参照**


- 「メトリック」1262 ページ
- 「警告スレッショルド」1305 ページ


## リソースからの未送信のエラーレポートに関する警告の抑制

♦ **未送信のエラーレポートに関する警告を抑制するには、次の手順に従います。**

管理者は、リソースに未送信のエラーレポートがある場合に警告を送信するかどうかを設定できます。デフォルトでは、このような警告は送信されません。エラーレポートとその送信方法の詳細については、「SQL Anywhere のエラーレポート」1210 ページを参照してください。

1. モニターに管理者としてログインします。
2. **[ツール]** » **[管理]** をクリックします。
3. **[設定]** をクリックします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

4. **[編集]** をクリックします。
5. **[オプション]** をクリックします。
6. **[リソースの未送信のエラーレポートの警告を表示しません。]** をクリックします。
7. **[保存]** をクリックします。
8. **[閉じる]** をクリックします。

## 警告スレッショルド

管理者は、警告のトリガーに使用するスレッショルドを設定できます。「警告スレッショルドの指定」1304 ページを参照してください。

警告とそのデフォルトのスレッショルドを次に示します。

同期システムはそれぞれ動作と制約が異なるので、デフォルト設定が環境に不適切である場合があります。各警告スレッショルドを確認し、必要に応じて調整してください。

### SQL Anywhere データベースリソースの警告スレッショルド

- **[CPU 使用率が、指定した秒数の間に指定したスレッショルドを超えた場合に警告します]**
  - **[スレッショルド (%)]**　リソースの CPU 使用率が指定された割合に達すると、警告が発行されます。デフォルトは 95% です。
  - **[秒]**　デフォルトは 30 秒です。
- **[メモリ使用率が最大キャッシュサイズの次の割合に達したときに警告します : *X*%]**　リソースのメモリ使用率が指定された割合に達すると、警告が発行されます。デフォルトは 90% です。
- **[DB 領域あたりの空きディスク領域が次の値を下回ったときに警告します : *X* MB]**　DB 領域あたりの空きディスク領域が指定された容量より少なくなると、警告が発行されます。デフォルトは 100 MB です。
- **[接続がブロックされた時間が次の時間を超えたときに警告します : X 秒]**　接続がブロックされた時間が指定された時間を超えると、警告が発行されます。デフォルトは 10 秒です。
- **[使用接続数がライセンス制限の次の割合に達したときに警告します : *X* %]**　使用中の接続数がライセンス制限に対して指定された割合に達すると、警告が発行されます。デフォルトは 85% です。
- **[クエリの実行時間が次の時間を超えたときに警告します : *X* 秒]**　クエリの実行時間が指定された時間を超えると、警告が発行されます。デフォルトは 10 秒です。
- **[スケジュールされていない要求の数が次の値に達したときに警告します : *X*]**　スケジュールされていない要求の数が指定された数に達すると、警告が発行されます。デフォルトは 5 です。

- **[最後のサーバー側のバックアップから指定日数を経過したときに警告します]**　リソースデータベースが最後にサーバー側でバックアップされてからの日数が指定した日数を超えると、警告を発行します。デフォルトは 14 日です。

- **[切断されたスケールアウトノードの数が指定したスレッショルドを超えたときに警告します :]**　読み込み専用スケールアウトシステムまたはミラーリングシステムにデータベースが含まれていて、システムで、切断されたノードが指定の率になっているときに、警告を発行します。

- **[同じ状況が次の時間内で発生した場合は警告しません : *X* 分]**　このオプションにより、指定した時間内に警告を重複して受信することがなくなります。デフォルトは 30 分です。

**Mobile Link サーバーリソースの警告スレッショルド**

- **[CPU 使用率が、指定した秒数の間に指定したスレッショルドを超えた場合に警告します]**　**[スレッショルド]** のデフォルトは 100% です。**[秒]** のデフォルトは 300 です。

- **[使用されるキャッシュページの割合が次の値 (%) よりも大きい場合に警告します]**　デフォルトは 100 です。

- **[ロックされているキャッシュページの割合が次の値 (%) よりも大きい場合に警告します]**　デフォルトは 80 です。

- **[1 秒間にスワップインおよびスワップアウトされているページ数が、指定した秒数の間に指定したスレッショルドを超えた場合に警告します]**　[スレッショルド] のデフォルトは 256 です。[秒] のデフォルトは 120 です。

- **[最も長いアクティブな同期の時間が次の値 (秒) よりも長い場合に警告します]**　デフォルトは 600 です。

- **[失敗した同期の数が、指定した分数の間に指定したスレッショルドを超えた場合に警告します]**　失敗した同期の [スレッショルド] のデフォルトは 20 です。[分] のデフォルトは 60 です。

- **[エラー数が、指定した分数の間に指定したスレッショルドを超えた場合に警告します]**　[スレッショルド (エラー数)] のデフォルトは 50 です。[分] のデフォルトは 60 です。

- **[Mobile Link キャッシュ用ディスク空き領域が次の値 (MB) を下回ったときに警告します]**　デフォルトは 100 です。

- **[データベースワーカースレッドの最も長いアクティブ待機が次の値 (秒) よりも長い場合に警告します]**　デフォルトは 300 です。

- **[同じ状況が次の時間内で発生した場合は警告しません]**　このオプションにより、指定した時間内に警告を重複して受信することがなくなります。デフォルトは 30 分です。

**Relay Server ファームリソースの警告スレッショルド**

- **[Relay Server の CPU 使用率が、指定した秒数の間に指定したスレッショルドを超えた場合に警告します]**　**[スレッショルド]** のデフォルトは 80% です。**[秒]** のデフォルトは 30 です。

- **[バックエンドファームの接続が部分的にのみ使用できる時間が指定秒数を超えた場合に警告します]** デフォルトは 30 です。
- **[Relay Server の共有メモリ使用率が、指定した秒数の間に指定したスレッショルドを超えた場合に警告します]** **[スレッショルド]** のデフォルトは 80 です。**[秒]** のデフォルトは 30 です。
- **[バックエンドファームが 2 分以上使用できない場合に警告します。]** このオプションはデフォルトで選択解除されています。
- **[Relay Server のファームで致命的なエラーが発生した場合に警告します。]** このオプションはデフォルトで選択されています。
- **[同じ状況が次の時間内で発生した場合は警告しません]** このオプションにより、指定した時間内に警告を重複して受信することがなくなります。デフォルトは 30 分です。

## 警告の電子メール送信

**♦ モニターによって警告の通知が電子メールで送信されるようにするには、次の手順に従います。**

管理者は、指定したリソースに対する警告が発生したときに、指定したオペレーターや管理者に電子メールが送信されるようモニターを設定できます。

1. モニターに管理者としてログインします。
2. 電子メールアドレスを持つ管理者またはオペレーターを作成します。「モニターユーザーの作成」1298 ページを参照してください。
3. 作成した管理者またはオペレーターにリソースを関連付けます。「モニターユーザーとリソースの関連付け」1299 ページを参照してください。
4. モニターでの電子メールの送信を有効にします。「モニターによる警告電子メールの送信の有効化」1307 ページを参照してください。

### モニターによる警告電子メールの送信の有効化

**♦ モニターによって警告の通知が電子メールで送信されるようにするには、次の手順に従います。**

管理者は、警告が発生したときにオペレーターや管理者に電子メールが送信されるようモニターを設定できます。モニターでは、SMTP プロトコルと MAPI プロトコルを使用した電子メール送信をサポートしています。

1. モニターに管理者としてログインします。
2. **[ツール]** » **[管理]** をクリックします。
3. **[設定]** をクリックします。

4. **[編集]** をクリックします。
5. **[警告の通知]** をクリックします。
6. **[警告の通知を電子メールで送信]** をクリックします。
7. **[電子メールでの警告の送信に使用するプロトコルを指定してください。]** フィールドで **[SMTP]** または **[MAPI]** をクリックします。
8. その他の設定を必要に応じて指定します。
   - **MAPI**
     - **[ユーザー名]**　MAPI サーバーのユーザー名を入力します。
     - **[パスワード]**　MAPI サーバーのパスワードを入力します。
   - **[SMTP]**
     - **[サーバー]**　使用する SMTP サーバーを指定します。SMTP サーバーのサーバー名か IP アドレスを入力します。たとえば、*SMTP.yourcompany.com* のように入力します。
     - **[ポート]**　SMTP サーバーの接続ポート番号を指定します。デフォルトは 25 です。
     - **[送信者名]**　送信者の電子メールアドレスのエイリアスを指定します。たとえば、*JoeSmith* のように指定します。
     - **[送信者のアドレス]**　送信者の電子メールアドレスを指定します。たとえば、*jsmith@emailaddress.com* のように指定します。
     - **[この SMTP サーバーは認証が必要]**　使用する SMTP サーバーに認証が必要な場合は、このオプションを選択します。
       - **[ユーザー名]**　認証が必要な SMTP サーバーに対して使用するユーザー名を指定します。
       - **[パスワード]**　認証が必要な SMTP サーバーに対して使用するパスワードを指定します。
     - **[この SMTP サーバーはセキュア (TLS) 接続が必要です]**　使用する SMTP サーバーに TLS 接続が必要な場合は、このオプションを選択します。
       - **[信頼できる証明書]**　SMTP サーバーでモニターが認証に使用する証明書ファイルのパスとファイル名を入力します。
       - **[証明書に記載される会社]**　証明書を発行した認証局の名前を入力します。SMTP サーバー側とモニター側の値を一致させる必要があります。
       - **[証明書に記載される部署]**　証明書に記載される部署を入力します。これは組織単位とも呼ばれます。SMTP サーバー側とモニター側の値を一致させる必要があります。
       - **[証明書に記載される名前]**　証明書の通称を入力します。SMTP サーバー側とモニター側の値を一致させる必要があります。
   - **[警告の通知] オプション**
     - **優先度の高い警告の場合のみ電子メール通知を送る**　優先度の高い警告が発行された場合のみ、電子メール通知を受け取るときは、このオプションを選択します。優先度が


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

中程度か低い警告は、引き続き **[警告リスト]** ウィジェットで表示して解決することができます。

○ **1 日に次に指定する数を超える電子メール通知を送信しない**　このオプションを選択すると、1 日当たりで各ユーザーに送信される電子メール通知の数が制限されます。**[電子メール通知]** フィールドに通知の最大数を設定します。デフォルト値は 25 です。

ユーザーが通知の最大数まで受け取ると、ユーザーに通知制限をリセットする方法を知らせる電子メールが送信されます。この制限は、真夜中に自動的にリセットされ、そのときにオペレーターは再度電子メール通知の受け取りを開始します。また、オペレーターは、**[警告の通知]** ウィンドウで **[電子メール通知の送信カウントをリセット]** ボタンをクリックすることでも、手動で制限をリセットできます。

9. **[テスト電子メールを送信]** をクリックします。

10. 画面の要求に従い、テスト電子メールの送信先電子メールアドレスを入力して、**[OK]** をクリックします。

    テスト電子メールが、指定された電子メールアドレスに送信されます。

11. **[保存]** をクリックします。

12. **[閉じる]** をクリックします。

    警告が発生すると、ユーザーアカウントに電子メールアドレスを指定されているオペレーターと管理者に警告の電子メールが送信されます。これらのユーザーは、関連付けられたリソースの電子メールを受け取ります。「モニターユーザーの作成」1298 ページと「モニターユーザーとリソースの関連付け」1299 ページを参照してください。

**参照**


- 「警告の解決」1302 ページ
- 「警告の削除」1303 ページ
- 「警告の表示」1302 ページ
- 「xp_startsmtp システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』


# モニターのバックアップ

モニターでは、デフォルトで、毎日 1 回夜中の 12 時にメトリックのメンテナンスが実行されます。メンテナンスの対象はメトリックで、警告は対象ではありません。

管理者は、次の操作を実行できます。

- モニターにメトリックをバックアップするスケジュールを設定する。
- モニターで使用されるディスク領域を制御する。
- メンテナンスをオンデマンドで実行する。

**♦ メトリックをバックアップし、メトリックの格納に使用されるディスク領域を制御するには、次の手順に従います。**

1. モニターに管理者としてログインします。
2. **[管理]** をクリックします。
3. **[設定]** をクリックし、**[編集]** をクリックします。
4. **[メンテナンス]** をクリックします。
5. メンテナンスの実行時刻 (24 時間表示) を指定します。デフォルトでは、夜中の 12 時に実行されます。この時刻は、モニターが実行されているコンピューターのローカル時間に対応します。「時刻の表示方法の概要」1247 ページを参照してください。
6. バックアップデータを保存するフォルダーを指定します。フォルダーは、モニターが実行されているコンピューターに存在する必要があります。
7. **[データの削減]** 設定をカスタマイズします。
   - **[次より古いメトリックは、1 日ごとに値をまとめる]**　このオプションを選択すると、指定された日数より古いすべての数値メトリックの平均が計算され、その数値メトリックが削除されます。非数値メトリックは削除されません。
   - **[次よりも古い値を削除する]**　このオプションを選択すると、指定された日数より古いすべてのメトリックが削除されます。
   - **[SQL Anywhere モニターによって使用されている総ディスク領域が次の容量 (MB) を超えた場合に古いメトリックを削除する]**　このオプションを選択した場合は、メトリックの保存に使用できる容量の上限を指定します。使用されているディスク領域がこの指定値以上になると、メトリックが古い順に削除されます。メトリックは、新しいメトリックを保存するのに十分な空きディスク領域が確保されるまで削除されます。
8. **[メンテナンスを今すぐ実行]** をクリックして、すぐにバックアップを実行します。プロンプトが表示されたら、**[保存]** をクリックします。
9. **[OK]** をクリックします。
10. **[Cancel]** をクリックします。

次の手順では、モニターデータベースをバックアップコピーに置き換える方法について説明します。

**♦ モニターデータベースのバックアップコピーをリストアするには、次の手順に従います。**

1. モニターが実行されている場合は、停止します。
2. バックアップフォルダーから *samonitor.db* データベースファイルと *samonitor.log* ログファイルをコピーします。
3. 現在のモニターデータベースファイルとログファイルが格納されているフォルダーにこれらのファイルを貼り付けます。プロンプトが表示されたら、既存のファイルを上書きします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

バージョン 12.0.1 モニターデータベースファイルのデフォルトのロケーションを次の表に示します。

| オペレーティングシステム | モニターのフォルダー |
|---|---|
| Windows XP (SQL Anywhere とともにインストール) | C:¥Documents and Settings¥All Users ¥Documents¥SQL Anywhere 12¥Monitor ¥samonitor.db |
| Windows XP (別のコンピューターにインストール) | C:¥Documents and Settings¥All Users ¥Documents¥SQL Anywhere 12 Monitor ¥samonitor.db |
| Windows Vista (SQL Anywhere とともにインストール) | C:¥Users¥Public¥Documents¥SQL Anywhere 12¥Monitor¥samonitor.db |
| Windows Vista (別のコンピューターにインストール) | C:¥Users¥Public¥Documents¥SQL Anywhere 12 Monitor¥samonitor.db |
| Linux (SQL Anywhere とともにインストール) | /opt/sqlanywhere12/samonitor.db |
| Linux (別のコンピューターにインストール) | /opt/sqlanywhere12/samonitor.db |

4. モニターを再起動します。

## モニターがインストールしたデータベースオブジェクトのリスト

次の表に、SQL Anywhere データベースをモニタリング対象のリソースとして追加する場合にインストールされるオブジェクトを示します。

| オブジェクト名 | オブジェクトタイプ | 説明 |
|---|---|---|
| sa_monitor_user | データベースユーザー | これは読み込み専用ユーザーで、メトリックを収集するためにデータベースに追加されます。このユーザーはモニタリングされるデータベースに追加されるので、そのデータベース外部のどこかに DBA クレデンシャルを格納する必要はありません。sa_monitor_user に対して、パスワード検証の回避を許可する必要が生じる場合があります。sa_monitor_user は、モニターだけが知っているランダムなパスワードを持ち、管理者権限は持っていません。 |

| オブジェクト名 | オブジェクトタイプ | 説明 |
|---|---|---|
| sa_monitor_connection_failure | テーブル | このテーブルには、失敗した接続試行に関するメトリックが含まれ、sa_monitor_connection_failed_event と共に使用されます。このテーブルに含まれるメトリックは、メトリックがモニターによって取得されると削除されます。 |
| sa_monitor_connection_failed_event | イベント | このイベントは、ConnectFailed システムイベントを接続試行が失敗するたびに発行し、レコードを sa_monitor_connection_failure テーブルに挿入します。 |
| sa_monitor_count_unsubmitted_crash_reports | 関数 | この関数は、xp_srvmon_count_unsubmitted_crash_reports プロシージャーを呼び出し、未送信のクラッシュレポートの数を判断します。 |

### モニタリングオブジェクトの削除

データベースオブジェクトの所有者は 1 人なので、次の文を実行することですべてのオブジェクトを削除できます。

```
DROP USER sa_monitor_user;
```

### モニタリングオブジェクトの再インストール

データベースオブジェクトを再インストールする方法については、「データベースリソースの修復」1260 ページを参照してください。

## 運用環境での SQL Anywhere モニターの実行

運用環境では、特に次のことをおすすめします。

1. モニター Production Edition をインストールし、実行します。

   モニター Production Edition を起動することには、次の利点があります。

   - モニターがサービスとしてバックグラウンドで実行される。
   - コンピューターの起動時にモニターが自動的に起動される。
   - SQL Anywhere のアップグレードや更新によって、モニター Production Edition が上書きされたり、影響を受けたりすることはない。これに対し、モニター Developer Edition は、SQL Anywhere のアップグレードや更新によって影響を受ける場合があります。これは、Developer Edition では、インストールされた SQL Anywhere をバックエンドで使用するためです。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

2. リソースが実行されているコンピューターとは別のコンピューターにモニターをインストールします。

   モニターを別のコンピューターで実行することには、次の利点があります。

   ●データベースサーバー、Mobile Link サーバー、Relay Server ファームやその他のアプリケーションに対する影響が最小限に抑えられます。

   ●リソースがインストールされているコンピューターに問題が発生しても、モニタリングは影響を受けません。

### SQL Anywhere Monitor Production Edition のインストール

**♦ モニター Production Edition をインストールするには、次の手順に従います (Windows の場合)。**

1. インストールメディアの *Monitor* ディレクトリから *setup.exe* を実行して、提供される指示に従います。

   Windows では、モニターサービスはインストールによって起動されます。

2. モニターにログインするために、デフォルト URL (*http://localhost:4950*) を開きます。

   **注意**
   ネットワークを経由してモニターにアクセスしている場合は、**http:**//*computer-name***:4950** をブラウズします。*computer-name* は、モニターが実行されているコンピューターの名前です。「モニターにリモートでログイン」1239 ページを参照してください。

3. ログインします。

   要求された場合は、モニターのユーザー名とパスワードを入力します。デフォルトのユーザーは、名前 **admin** とパスワード **admin** を持つ管理者です。「モニターのユーザー」1296 ページを参照してください。

**♦ モニター Production Edition をインストールするには、次の手順に従います (Linux の場合)。**

1. root ユーザーでインストールメディアの *Monitor* ディレクトリから *setup.tar* を実行し、表示される指示に従います。

   **注意**
   Linux では、モニター Production Edition を実行できるのは root ユーザーのみです。

2. Linux では、デフォルトで、モニター Production Edition はモニターサービスを自動的に起動します。

3. モニターにログインするために、デフォルト URL (*http://localhost:4950*) を開きます。

**注意**
ネットワークを経由してモニターにアクセスしている場合は、**http://***computer-name***:4950** をブラウズします。*computer-name* は、モニターが実行されているコンピューターの名前です。「モニターにリモートでログイン」1239 ページを参照してください。

4. ログインします。

   要求された場合は、モニターのユーザー名とパスワードを入力します。デフォルトのユーザーは、名前 **admin** とパスワード **admin** を持つ管理者です。「モニターのユーザー」1296 ページを参照してください。

# モニターのアップグレードおよびリソースとメトリックの移行

**警告**
モニターをアンインストールすると、アプリケーションだけでなく、リソース、収集したメトリックも削除されます。

現在のモニターのリソースとメトリックを保持する場合は、次の操作を実行します。

1. モニターの新しいバージョンをインストールします。
2. リソースとメトリックを移行します。
3. モニターの古いバージョンをアンインストールします。

「SQL Anywhere モニターのアップグレードおよびリソースとメトリックの移行」『SQL Anywhere 12 変更点とアップグレード』を参照してください。

## リソースと設定の移行

1 つのモニターから、新しくインストールしたモニターへリソースとメトリックを移行するには、移行ユーティリティを使用します。

**♦ 移行ユーティリティを使用したモニターリソースとデータの移行**

1 つのモニターから、**新しく**作成したモニターへリソースとメトリックを移行するには、移行ユーティリティを使用します。

**警告**
モニターをアンインストールすると、アプリケーションだけでなく、リソース、収集したメトリックも削除されます。

現在のモニターのリソースとメトリックを保持する場合は、次の操作を実行します。

1. モニターの新しいバージョンをインストールします。
2. リソースとメトリックを移行します。
3. モニターの古いバージョンをアンインストールします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

1. 既存のモニターデータベースファイル *samonitor.db* をコピーします。たとえば、11.0.1 モニターに移行するには、次のファイルをコピーします。

   C:¥Documents and Settings¥All Users¥Documents¥SQL Anywhere 11¥Monitor¥samonitor.db

2. 新しいモニターをインストールします。インストールメディアの *Monitor* ディレクトリにある *setup.exe* ファイルを実行し、表示される指示に従います。インストールが完了したら、新しいモニターを停止します (稼働している場合)。

   **注意**
   コンピューター上で一度に実行できるモニターのバージョンは 1 つだけです。

   バージョン 11.0.1 のモニターが稼働しているコンピューターにバージョン 12.0.1 のモニターをインストールすると、バージョン 11.0.1 のモニターが停止します。

3. コマンドプロンプトで移行ユーティリティを実行します。次の表を使用して、移行ユーティリティのロケーションを確認します。

| オペレーティングシステム | モニタータイプ | 移行ユーティリティのデフォルトロケーション |
|---|---|---|
| Windows | モニター Developer Edition | *C:¥Program Files¥SQL Anywhere 12¥Bin32¥run_migrator.cmd* |
| | モニター Production Edition | *C:¥Program Files¥SQL Anywhere 12¥Bin32¥run_migrator.cmd* |
| Linux | モニター Developer Edition | */opt/sqlanywhere12/bin32/run_migrator.sh* |
| | モニター Production Edition | */opt/samonitor12/bin32/run_migrator.sh* または */opt/samonitor12/bin64/run_migrator.sh* |

   次のオプションを指定して移行ユーティリティを実行します。

   - **-c**　リソースと構成設定のみを移行することを指定します。デフォルトでは、リソースと収集されたデータの両方が移行されます。
   - **-t temporary-directory**　テンポラリファイルのディレクトリを指定します。デフォルトでは、テンポラリファイルは、run_migrator ファイルと同じディレクトリに作成されます。

   **注意**
   モニター移行で作成されるテンポラリファイルは、移行プロセスの最後で削除されます。-t オプションを使用して、これらのテンポラリファイルのディレクトリを指定します。テンポラリファイルには、バージョン 11.0.1 のモニターデータベースファイルと同様の領域が必要です。指定のディレクトリに十分な領域があることを確認してください。

- **source-filename**　データのアンロード元となる古いモニターファイルのパスとファイル名を指定します。たとえば、バージョン 11.0.1 の *samonitor.db* ファイルのパスを指定します。
- ***destination-filename***　データの再ロード先となる新しいモニターファイルのパスとファイル名を指定します。たとえば、バージョン 12.0.1 の *samonitor.db* ファイルのパスを指定します。

次に例を示します。

```
C:¥Program Files¥SQL Anywhere 12¥run_migrator.cmd -t c:¥monitorbackup c:¥Program Files
¥SQL Anywhere 11¥Monitor¥samonitor11.db C:¥Program Files¥SQL Anywhere 12¥Monitor
¥samonitor12.db
```

バージョン 11.0.1 モニターおよびバージョン 12.0.1 モニターのデータベースファイルのデフォルトのロケーションを次の表に示します。

| オペレーティングシステム | モニタータイプ | 11.0.1 ディレクトリ | 12.0.1 ディレクトリ |
|---|---|---|---|
| Windows XP | モニター Developer Edition | C:¥Documents and Settings¥All Users ¥Documents¥SQL Anywhere 11¥Monitor ¥samonitor.db | C:¥Documents and Settings¥All Users ¥Documents¥SQL Anywhere 12¥Monitor ¥samonitor.db |
| Windows XP | モニター Production Edition | C:¥Documents and Settings¥All Users ¥Documents¥SQL Anywhere 11 Monitor ¥samonitor.db | C:¥Documents and Settings¥All Users ¥Documents¥SQL Anywhere 12 Monitor ¥samonitor.db |
| Windows Vista 以降のバージョンの Windows | モニター Developer Edition | C:¥Users¥Public ¥Documents¥SQL Anywhere 11¥Monitor ¥samonitor.db | C:¥Users¥Public ¥Documents¥SQL Anywhere 12¥Monitor ¥samonitor.db |
| Windows Vista 以降のバージョンの Windows | モニター Production Edition | C:¥Users¥Public ¥Documents¥SQL Anywhere 11 Monitor ¥samonitor.db | C:¥Users¥Public ¥Documents¥SQL Anywhere 12 Monitor ¥samonitor.db |
| Linux | モニター Developer Edition | /opt/sqlanywhere11/ samonitor.db | /opt/sqlanywhere12/ samonitor.db |
| Linux | モニター Production Edition | /opt/samonitor11/ samonitor.db | /opt/sqlanywhere12/ samonitor.db |

# モニター通信のセキュリティ

モニターとブラウザー間、およびモニターとモニタリング対象のリソース間の通信を保護できます。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### トランスポートレイヤーセキュリティ (TLS) を使用したモニターとブラウザー間の通信の保護

トランスポートレイヤーセキュリティ (TLS) を使用すると、モニターとブラウザー間の通信を保護することができます。モニターでは、SSL バージョン 3.0 と TLS バージョン 1.0 を使用した HTTPS 接続をサポートする Web サーバーが実行されます。

**♦ モニターに TLS セキュリティを設定するには、次の手順に従います。**

1. 認証局からデジタル証明書を取得するか、証明書の作成ユーティリティ (createcert) を使用して自己署名証明書を作成します。「証明書作成ユーティリティ (createcert)」924 ページを参照してください。

2. モニターの開始行の文字列を変更して、証明書を使用するようにします。

3. 必要に応じて、新しい証明書を受け入れるようにブラウザーを設定します。

詳細については、http://www.ianywhere.jp/tech/1063938-SecuringtheSAMonitorWithTransportLayerSecurity.html を参照してください。

### モニターとリソース間の通信の保護

ECC または FIPS 認定の暗号化を使用すると、モニターとモニタリング対象のリソース間の通信を暗号化できます。ECC 暗号化と FIPS 認定の暗号化には、別途ライセンスが必要です。

**注意**
別途ライセンスが必要な必須コンポーネント

ECC 暗号化と FIPS 認定の暗号化には、別途ライセンスが必要です。強力な暗号化テクノロジはすべて、輸出規制対象品目です。

「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

## モニターのトラブルシューティング

次に示す推奨事項に加えて、管理者はメッセージログ機能と例外レポート機能を使用して、モニターをトラブルシューティングできます。「メッセージログの表示。」1248 ページと「例外レポートの表示」1248 ページを参照してください。

| 問題 | 推奨事項 |
|---|---|
| [F5] キーを押してブラウザーウィンドウを再表示すると、SQL Anywhere モニターへのログインが要求される。 | ブラウザーで JavaScript を有効にします。 |
| モニターにログインしようとすると、ネットワーク通信エラーになる。 | モニターを起動します。「モニターの起動」1235 ページを参照してください。 |

| 問題 | 推奨事項 |
| --- | --- |
| Adobe Flash Player を最新バージョンにアップグレードした後も Adobe Flash Player のアップグレードを求めるメッセージが表示される。 | インストールされている Adobe Flash Player のバージョンが、オペレーティングシステムでサポートされていることを確認します。モニターには、Adobe Flash Player のバージョン 10 との下位互換性があります。適切なバージョンを確認するには、次の URL を参照してください。 |
| モニターで SQL Anywhere データベースリソースのモニタリングを開始できない。 | 該当リソースのパスワード検証関数とログインプロシージャーで、user sa_monitor_user によるリソースへの接続が許可されていることを確認します。 |
| 警告の電子メールをまったく受信しない。 | モニターが電子メールを送信するよう適切に設定されていることを確認し、テスト電子メールを送信します。「モニターによる警告電子メールの送信の有効化」1307 ページを参照してください。<br><br>モニターが送信する警告の電子メールがウイルススキャナーによってブロックされていないことを確認します。「xp_startsmtp システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。 |
| モニターからレポートされるスケジュールされていない要求の数が、実際の数より少ない。 | スケジュールされていない要求の数に関するメトリックを収集する際、モニターはリソースに対してクエリを実行します。このクエリがスケジュールされていない要求になっている可能性が考えられます。<br><br>スケジュールされていないクエリは、到着順に処理されます。そのため、モニターがクエリを実行しようとしたときにスケジュールされていない要求が存在すると、そのクエリは既存のスケジュールされていない要求が完了するのを待って処理されます。<br><br>その結果、モニターがスケジュールされていない要求の数を収集するとき、その数には、モニターがクエリを発行してからそのクエリが実行されるまでに存在したスケジュールされていない要求が含まれません。 |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| 問題 | 推奨事項 |
| --- | --- |
| データベースのディスク領域が指定されたスレッショルドを超えたのに警告を受信しない。 | データベースは、モニターによる収集間隔の間に、指定されたディスク領域の警告スレッショルドと使用可能な領域の上限を超える可能性があります。このような状況になると、モニターがディスク使用率メトリックを収集して警告を発行する前に、データベースが応答しなくなります。<br><br>データベースのサイズが急激に増大する場合は、ディスク領域の警告スレッショルドの値を大きくして、データベースが領域不足になる前に警告を受信できるようにします。「警告スレッショルド」1305 ページを参照してください。 |
| モニターにログインしているとき、**[管理]** ウィンドウを表示できない。 | **[管理]** ウィンドウにアクセスするには、モニターに管理者としてログインする必要があります。「モニターのユーザー」1296 ページを参照してください。 |
| データを移行する前にモニターをアンインストールし、リソースとメトリックのデータが失われた。 | モニターをアンインストールすると、アプリケーションだけでなく、リソース、収集したメトリックも削除されます。アップグレードするときは、モニターの新しいバージョンをインストールし、データを移行してから、古いバージョンをアンインストールする必要があります。<br><br>ただし、モニターを定期的にバックアップしている場合は、移行ユーティリティを使用して、バックアップファイルからモニターの新しいバージョンへデータを移行できます。「モニターのバックアップ」1309 ページと「モニターのアップグレードおよびリソースとメトリックの移行」1314 ページを参照してください。 |

# SQL Anywhere SNMP Extension Agent

SQL Anywhere を Windows (32 ビットバージョン) 上で実行している場合は、SQL Anywhere SNMP Extension Agent と SNMP 管理アプリケーションを組み合わせて SQL Anywhere データベースを管理することができます。異なるコンピューターで動作している複数のデータベースサーバー上の各データベースを 1 つのエージェントでモニタリングできます。

SQL Anywhere SNMP Extension Agent を使用すると、次の操作を実行できます。

●個々のサーバー統計とデータベース統計の値を検索する。

●個々のサーバープロパティとデータベースプロパティの値を検索する。

●個々の PUBLIC データベースオプションの値を検索する。

●いずれかの PUBLIC データベースオプションに値を設定する。

●ストアドプロシージャーを実行する。

●プロパティ値または統計値に基づいてトラップを生成する。

### 提供されるファイル

SQL Anywhere インストール環境には、以下の SQL Anywhere SNMP Extension Agent 用ファイルが用意されています。

- ***dbsnmp12.dll*** SQL Anywhere SNMP Extension Agent。このファイルは、*%SQLANY12%¥bin32* にあります。
- ***iAnywhere.mib*** SQL Anywhere MIB には、データベースサーバーとデータベースのプロパティ、統計、オプションのうち SQL Anywhere SNMP Extension Agent を使用してアクセスできるものの OID が、すべて格納されています。
- ***RDBMS-MIB.mib*** これはリレーショナルデータベース管理システムのための汎用 MIB であり、SQL Anywhere SNMP Extension Agent を使用してアクセスできる OID が格納されています。
- ***SNMPv2-SMI.mib*** この MIB は、Adaptive Server Anywhere MIB と RDBMS MIB から参照されます。
- ***SNMPv2-TC.mib*** この MIB は、Adaptive Server Anywhere MIB と RDBMS MIB から参照されます。
- ***SYBASE-MIB.mib*** Sybase MIB です。この MIB は、ASQL Anywhere MIB から参照されます。
- ***sasnmp.ini*** このファイルには、SQL Anywhere SNMP Extension Agent によるモニタリングの対象となるデータベースが列記されています。デフォルトで、このファイルは *%SQLANY12%¥bin32* にあります。

## SNMP

Simple Network Management Protocol (**SNMP**) は、ネットワーク管理に使用されている標準的なプロトコルです。SNMP では、「マネージャー」と「エージェント」が通信できます。マネージャーはエージェントに要求を送り、エージェントはマネージャーからのクエリに応答します。エージェントは、特定のイベントが発生した場合に「トラップ」と呼ばれる通知を使用してマネージャーに通知することもできます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

SNMP エージェントは、管理対象オブジェクトに対する変数値の取得要求と設定要求を処理します。各変数には値が 1 つあります。この値は一般に文字列型または整数型ですが、他のデータ型の場合もあります。

変数は、1 つのグローバルな階層に保持されます。各変数には、その親の下でユニークとなる番号が割り当てられます。変数の完全名 (すべての親を含めた名前) は、「オブジェクト識別子」(OID) と呼ばれます。Sybase 固有の OID は、いずれも **1.3.6.1.4.1.897** から始まります。

エージェントがサポートする OID のリストは、「Management Information Base」 (MIB) と呼ばれるファイルに保存されています。このリストには、OID の名前や型などの情報も含まれています。

MIB は、管理対象オブジェクトについてのネットワーク管理情報が格納されているデータベースです。MIB は、SQL Anywhere SNMP Extension Agent を使用してモニタリングする SQL Anywhere データベースとは別のものです。MIB オブジェクトの値は、SNMP を使用して変更または検索できます。MIB オブジェクトは 1 つの階層にまとめられており、この階層の最上位には、ネットワークについての最も一般的な情報が置かれています。SQL Anywhere SNMP Extension Agent は、以下の MIB をサポートします。

- **SQL Anywhere MIB**　SQL Anywhere SNMP Extension Agent 専用の MIB。SQL Anywhere MIB 内の OID は、いずれも **1.3.6.1.4.1.897.2** から始まります。SQL Anywhere MIB には、SQL Anywhere SNMP Extension Agent を使用して検索 (または検索と設定の両方) ができる統計、プロパティ、オプション値の OID が登録されています。「SQL Anywhere MIB」1321 ページを参照してください。

- **RDBMS MIB**　特定のベンダーに依存しない、リレーショナルデータベースのための汎用 MIB。この MIB には、システム内のデータベースサーバーとデータベースについての情報が格納されています。「RDBMS MIB」1324 ページを参照してください。

## SQL Anywhere MIB

SQL Anywhere MIB は、SQL Anywhere SNMP Extension Agent 用に作成されています。データベースサーバーのすべての統計とプロパティ、およびデータベースのすべての統計、プロパティ、オプションが格納されています。統計とプロパティはいくつかの例外を除いてすべて読み込み専用ですが、データベースのオプションはいずれも読み込みと書き込みの両方が可能です。

デフォルトで、SQL Anywhere MIB は *%SQLANY12%¥snmp¥iAnywhere.mib* にあります。

SQL Anywhere MIB 内のテーブルの詳細については、「SQL Anywhere MIB リファレンス」1333 ページを参照してください。

SQL Anywhere MIB 内の値を設定する方法の詳細については、「SQL Anywhere SNMP Extension Agent による値の設定」1329 ページを参照してください。

SQL Anywhere MIB の階層は、以下の説明のとおりです。

| OID | 名前 | 説明 |
|---|---|---|
| 1.3.6.1.4.897.2.1.1.*n*.*db* | saServer.saSrvStat | データベース *db* のサーバー統計 *n* の値を返す |
| 1.3.6.1.4.897.2.1.2.*n*.*db* | saServer.saSrvProp | データベース *db* のサーバープロパティ *n* の値を返す |
| 1.3.6.1.4.897.2.2.1.*n*.*db* | saDb.saDbStat | データベース *db* のデータベース統計 *n* の値を返す |
| 1.3.6.1.4.897.2.2.2.*n*.*db* | saDb.saDbProp | データベース *db* のデータベースプロパティ *n* の値を返す |
| 1.3.6.1.4.897.2.2.3.*n*.*db* | saDb.saDbOpt | データベース *db* のデータベースオプション *n* の値を返す |
| 1.3.6.1.4.897.2.3.1 | saAgent.saVersion | SQL Anywhere Extension Agent のバージョンを返す |
| 1.3.6.1.4.897.2.3.2.*db* | saAgent.saDbConnStr | データベース *db* 用の接続文字列を返す |
| 1.3.6.1.4.897.2.3.3.*db* | saAgent.saConnected | SQL Anywhere Extension Agent がデータベース *db* に接続されているかどうかを示す値を返す。0 を設定すると、SQL Anywhere Extension Agent はデータベースとの接続を切断します。1 を設定すると、SQL Anywhere Extension Agent はデータベースに接続します。 |
| 1.3.6.1.4.897.2.3.4.*db* | saAgent.saStarted | データベース *db* が動作しているかどうかを返す。0 を設定すると、SQL Anywhere Extension Agent はデータベースを停止します。[1] 1 を設定すると、データベースを起動しようとします。[2] |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| OID | 名前 | 説明 |
|---|---|---|
| 1.3.6.1.4.897.2.3.5.*db* | saAgent.saProc | 文字列 *proc_name* を設定すると、SQL Anywhere Extension Agent はデータベース内のプロシージャー *proc_name* を実行する。<br><br>引数は proc_name('string', 4) などの形式で指定できます。引数を指定しなかった場合には、名前の後に空のカッコ **()** が追加されます。値の取得を行った場合には、**""** が返されます。 |
| 1.3.6.1.4.897.2.3.6 | saAgent.saRestart | この変数の値を 1 に設定すると、エージェントが自動的に再起動する (すべてのデータベースとの接続を切断し、*.ini* ファイルを再ロードする)。値の取得を行った場合には、0 が返されます。 |
| 1.3.6.1.4.897.2.3.7 | saAgent.saInifile | SQL Anywhere Extension Agent が使用している *sasnmp.ini* ファイルのフルパスを返す |
| 1.3.6.1.4.897.2.4 | saMetaData | 複数の仮想テーブル。各ローが SQL Anywhere MIB でサポートされる変数の 1 つを示します。 |

[1] この変数の値を設定することによってデータベースを停止すると、無条件で停止が実行されます。つまり、アクティブな接続がある場合でもデータベースは停止します。

[2] この変数の値を設定することによってデータベースを起動する場合は、接続文字列内に DBF パラメーターを指定しておく必要があります (DBN と、必要な場合は DBKEY も含みます)。また、*sasnmp.ini* ファイル内に **UtilDbPwd** フィールドを設定するか、サーバーのデータベース起動パーミッション (-gd サーバーオプションを使用して指定します) を all に設定することが必要です。

### asaMetaData テーブル

SQL Anywhere MIB には、サポートされている変数を調べるときに SQL Anywhere Extension Agent への問い合わせの手段として利用できるメタデータテーブルが含まれています。

- **saSrvMetaData.saSrvStatMetaDataTable** データベースサーバー統計 (sa.saServer.saSrvStat に属する変数) を示します。

- **saSrvMetaData.saSrvPropMetaDataTable**　データベースサーバープロパティ (sa.saServer.saSrv.Prop に属する変数) を示します。
- **saDbMetaData.saDbStatMetaDataTable**　データベース統計 (sa.saDb.saDbStat に属する変数) を示します。
- **saDbMetaData.saDbpropMetaDataTable**　データベースプロパティ (sa.saDb.saDbProp に属する変数) を示します。
- **saDbMetaData.saDbOptMetaDataTable**　データベースオプション (sa.saDb.saDbOpt に属する変数) を示します。

SQL Anywhere MIB メタデータテーブルに保存されている情報の詳細については、「asaMetaData テーブル」1334 ページを参照してください。

## RDBMS MIB

RDBMS MIB は、特定のベンダーに依存しない、リレーショナルデータベース管理システム製品のための汎用 MIB (RFC 1697) です。RDBMS MIB は、「仮想テーブル」を使用してサーバーとデータベースについての情報を返します。ベース OID は **1.3.6.1.2.1.39** です。この MIB 内には 9 つの仮想テーブルがあります。SQL Anywhere SNMP Extension Agent は、これらの仮想テーブルのうちの 8 つをサポートします。

RDBMS MIB 内のテーブルの詳細については、「RDBMS MIB リファレンス」1372 ページを参照してください。

SQL Anywhere Extension Agent からは、RDBMS MIB 内のサポートされる変数に読み込み専用でアクセスできます。RDBMS MIB 内の変数を SQL Anywhere Extension Agent で書き込むことはできません。

仮想テーブルは、一定数の属性と、不特定の数のローで構成されています。テーブル内の要素は、GET 要求を使用して検索します。このとき、テーブルの OID にカラム番号とロー番号を追加したものを指定します。テーブル OID の後には、次のように 1 を追加する必要があります。

```
table.1.column.rownum
```

デフォルトで、RDBMS MIB は *%SQLANY12%¥snmp¥RDBMS-MIB.mib* にあります。

# SQL Anywhere SNMP Extension Agent

SQL Anywhere SNMP Extension Agent を使用するには、コンピューターに SNMP がインストールされている必要があります。また、SQL Anywhere SNMP Extension Agent を使用してモニタリングするデータベースについての情報が保存されている *sasnmp.ini* ファイルを作成する必要があります。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## SNMP のインストール

SQL Anywhere Extension Agent を使用するためには、コンピューターに SNMP をインストールする必要があります。デフォルトでは、SNMP は Windows にはインストールされません。

SNMP のインストールについては、オペレーティングシステムのマニュアルを参照してください。

SNMP のインストールが完了すると、SNMP Service と SNMP Trap Service サービスが動作している状態になります。

SQL Anywhere をインストールする前に SNMP をインストールした場合は、SNMP サービスが SQL Anywhere SNMP Extension Agent を検出できるようにするため、SNMP サービスをいったん停止し、再起動する必要があります。SQL Anywhere をインストールした後で SNMP をインストールした場合は、SNMP サービスが自動的に SQL Anywhere SNMP Extension Agent を検出します。

**♦ SNMP サービスを再起動するには、次の手順に従います (コマンドラインの場合)。**

1. 次のコマンドを実行します。

   ```
   net stop snmp
   ```

   この操作で SNMP サービスが停止します。

2. 次のコマンドを実行します。

   ```
   net start snmp
   ```

   この操作で SNMP サービスが起動します。

## SQL Anywhere SNMP Extension Agent の設定

SQL Anywhere Extension Agent では、1 つ以上のデータベースをモニタリングできます。モニタリング対象のデータベース情報は、次に示すフォーマットで *sasnmp.ini* に保存されています。

```
[SAAgent]
TrapPollTime=time-in-seconds

[DBn]
ConnStr=connection-string
UtilDbPwd=utility-database-password
CacheTime=time-in-seconds
DBSpaceCacheTime=time-in-seconds
Trapt=trap-information
Disabled=1 or 0
```

*sasnmp.ini* ファイルは、デフォルトで SQL Anywhere のインストール時に *%SQLANY12%¥bin32* ディレクトリに格納されます。

### SAAgent セクション

*sasnmp.ini* ファイルの SAAgent セクションには、SQL Anywhere Extension Agent に関する情報があります。TrapPollTime フィールドが不要な場合は、セクション全体を省略できます。

**TrapPollTime**　この値は、動的トラップのポーリング頻度を指定します (動的トラップが指定されている場合)。SQL Anywhere SNMP Extension Agent は、デフォルトでは 5 秒おきに値をポーリングします。この値を 0 に設定すると、動的トラップが無効になります。このフィールドはオプションです。

### DBn セクション

*sasnmp.ini* ファイルの各 **DB*n*** セクションには、特定のデータベースの情報、そのデータベースへの接続方法、そのデータベース用の動的トラップを記述します。このセクションのフィールドでは、大文字と小文字が区別されます。

*n* の値は、データベースを識別する番号です。この番号は、1 から開始する連番にする必要があります。不連続にすることはできません。たとえば、*sasnmp.ini* ファイルにエントリ [DB1]、[DB2]、[DB4] がある場合、エントリ [DB3] が欠如しているためエントリ [DB4] は無視されます。

- **ConnStr**　データベースへの接続に使用される接続文字列です。データベースに接続するには、情報を不足なく指定する必要があります。このフィールドは必須です。

  ○ODBC データソースを使用してデータベースに接続する場合、使用するデータソースはユーザーデータソースではなくシステムデータソースである必要があります。

  ○SNMP Extention Agent はサービスとして実行されるので、統合化ログインを使用する場合は SYSTEM アカウントにマッピングする必要があります。しかし、この処理を行うと、サービスとして実行されるものはすべてパスワードを入力することなくデータベースに接続できてしまいます。別の方法として、サービスを実行するためのアカウントを変更し、その後でそのアカウントの統合化ログインを作成することもできます。

  ○接続文字列の先頭には、文字列 ASTART=NO;IDLE=0;CON=SNMP;ASTOP=NO を付けます。この文字列に基づいて次の設定が行われます。

    ●SQL Anywhere SNMP Extension Agent がデータベースサーバーを自動的に起動するのを防ぐ

    ●アイドルタイムアウトを無効にする (SQL Anywhere SNMP Extension Agent は一定時間アイドル状態になる可能性があるため)

    ●識別が可能なように接続に名前を付ける

    ●SQL Anywhere SNMP Extension Agent が接続を停止するときにデータベースが停止しないようにする

  *sasnmp.ini* ファイル内の接続文字列にこれらの値を指定すると、デフォルトの設定が無効にされ、*sasnmp.ini* に指定した値が使用されます。

- **UtilDbPwd**　データベースを起動するように sa.agent.saStarted を設定すると、SQL Anywhere SNMP Extension Agent は、DBF パラメーター (データベースファイルの場所をデータベースサーバーに指定するためのパラメーター) を使用してデータベースに接続しようと

します。ただし、データベースの起動に必要なパーミッションが DBA である場合、サーバーは接続を許可しません (これはネットワークサーバーのデフォルト設定であり、パーソナルサーバー、ネットワークサーバーとも -gd dba オプションを使用してこのように設定できます)。

このようなサーバー上でデータベースを起動するためには、SQL Anywhere SNMP Extension Agent が同じサーバー上ですでに動作中のデータベースに DBA 権限を持つユーザーとして接続することが必要です。この条件は、ユーティリティデータベースに接続することで満たすことができます。ユーティリティデータベースのパスワード (-su サーバーオプションで指定します) を *sasnmp.ini* ファイル内で指定した場合、SQL Anywhere Extension Agent は、データベースを起動するために、そのデータベースと同じサーバー上のユーティリティデータベースに接続し、START DATABASE 文を実行してから接続を切断します。このフィールドはオプションです。

- **CacheTime**　データベースからデータを検索するときに、検索したデータを SQL Anywhere SNMP Extension Agent 内にキャッシュすることができます。これによって、それ以後に同じタイプのデータ (サーバープロパティやデータベース統計など) を検索するときには、データベースとのやりとりが不要になります。データをキャッシュすると、それ以降の検索ではデータをすばやく取得できますが、そのデータは最新の状態でない可能性があります。CacheTime フィールドは、キャッシュ時間を変更したり、キャッシュを無効にしたりする (値を 0 に設定すると無効になります) ために使用できます。デフォルトでは、キャッシュ時間は 0 秒です。CacheTime パラメーターを 0 に設定すると、取り出されるデータは常に最新となります。これは、要求ごとにデータがデータベースから検索されるためです。このフィールドはオプションです。

- **DBSpaceCacheTime**　RDBMS MIB 内の rdbmsDbLimitedResourceTable には、DB 領域についての情報が格納されています。この情報も、データベースから検索されたときに SQL Anywhere Extension Agent 内にキャッシュすることができます。DB 領域情報のデフォルトのキャッシュ時間は 600 秒 (10 分) です。このフィールドは、キャッシュ時間を変更するために使用できます。値を 0 に設定すれば、キャッシュを無効にすることもできます。このフィールドはオプションです。「rdbmsDbLimitedResourceTable」1374 ページを参照してください。

- **Trap *t***　動的トラップを作成します。値 *t* には正の整数を指定します。指定した値は、開始値が 1 の連番になっている必要があります。不連続にすることはできません。このフィールドはオプションです。「動的トラップ」1331 ページを参照してください。

- **Disabled**　SQL Anywhere SNMP Extension Agent は、このフィールドが 1 に設定されているデータベースエントリを無視します。このことを利用すると、SQL Anywhere SNMP Extension Agent の管理対象データベースのリストから特定のデータベースを一時的に削除する場合、他のデータベースの番号を変更する必要がありません。このフィールドはオプションです。

このファイルを編集した場合は、SNMP サービスを再起動するか、SQL Anywhere SNMP Extension Agent をリセットして、新しい設定が使用されるようにする必要があります。

### SNMP の再起動

**♦ SNMP サービスを再起動するには、次の手順に従います (コマンドラインの場合)。**

1. 次のコマンドを実行します。

```
net stop snmp
```

この操作で SNMP サービスが停止します。

2. 次のコマンドを実行します。

```
net start snmp
```

この操作で SNMP サービスが起動します。

**♦ SQL Anywhere SNMP Extension Agent の再起動**

● SNMP 管理ツールを使用して、saAgent.saRestart プロパティ (1.3.6.1.4.1.897.2.3.6) の値を **1** に変更します。

ファイル難読化ユーティリティ (dbfhide) を使用すると、単純暗号化によって *sasnmp.ini* ファイルの内容を難読化することができます。「.ini ファイルの内容の非表示」465 ページを参照してください。

### サンプル sasnmp.ini ファイル

SQL Anywhere SNMP Extension Agent の *sasnmp.ini* ファイルの例を以下に示します。

```
[SAAgent]
[DB1]
ConnStr=UID=DBA;PWD=sql;DBN=sales;DBF=sales.db
Trap1=1.1.5 > 50000
UtilDbPwd=test
[DB2]
ConnStr=UID=DBA;PWD=sql;DBN=field;DBF=field.db
UtilDbPwd=test
Disabled=1
[DB3]
ConnStr=UID=DBA;PWD=sql;Host=host2;DBN=hq;DBF=hq.db
UtilDbPwd=test
```

SAAgent セクションにパラメーターが指定されていないので、SQL Anywhere SNMP Extension Agent は 5 秒おきに値をポーリングします。

SQL Anywhere SNMP Extension Agent は、2 台のサーバーで動作している 3 つの異なるデータベースをモニタリングします。データベース 3 は別のコンピューターで動作しているため、Host 接続パラメーターが必要です。DB1 には、データベースサーバーから送信されたデータが 50000 バイトを超えている場合に起動するトラップが指定されています。

## SQL Anywhere SNMP Extension Agent による値の取得

SQL Anywhere SNMP Extension Agent を使用すると、次の値を検索できます。

●データベースサーバープロパティ ー 「SQL Anywhere MIB サーバープロパティ」1341 ページを参照してください。

●データベースサーバー統計 ー 「SQL Anywhere MIB サーバー統計」1337 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

●データベースオプション ー「SQL Anywhere MIB データベースオプション」1362 ページを参照してください。

●データベースプロパティ ー「SQL Anywhere MIB データベースプロパティ」1355 ページを参照してください。

●データベース統計 ー「SQL Anywhere MIB データベース統計」1350 ページを参照してください。

これらの値を取得する方法は、使用する SNMP 管理ソフトウェアによって異なります。

**例**

次の表は、いくつかの OID の説明と、返される値 (それぞれの OID の値) の例を示します。

| OID | 説明 | 値の例 |
|---|---|---|
| 1.3.6.1.4.1.897.2.1.1.1.1 | データベース 1 のサーバー統計 ActiveReq | 1 |
| 1.3.6.1.4.1.897.2.2.1.4.1 | データベース 1 のデータベース統計 CacheRead | 11397 |
| 1.3.6.1.4.1.897.2.3.1 | エージェントのバージョン | 12.0.1(2459) |
| 1.3.6.1.4.1.897.2.3.2.1 | データベース 1 用の接続文字列 | UID=DBA;PWD=sql; Host=host1; DBN=sales; DBF=sales.db |

## SQL Anywhere SNMP Extension Agent による値の設定

SQL Anywhere SNMP Extension Agent は、SNMP の get クエリ、get-next クエリ、set クエリに応答します。

SQL Anywhere SNMP Extension Agent を使用すると、任意のデータベースオプション、一部のサーバープロパティ、1 つのデータベースプロパティを設定できます。

SQL Anywhere SNMP Extension Agent は、データベースオプションの設定時に、次の文を実行します。

```
SET OPTION PUBLIC.option-name = 'value'
```

データベースプロパティとサーバープロパティの設定には、sa_server_option システムプロシージャーを使用します。

これらの値の設定方法は、使用する SNMP 管理ソフトウェアによって異なります。

SQL Anywhere SNMP Extension Agent を使用して設定できるオプションとプロパティの詳細については、「SQL Anywhere MIB リファレンス」1333 ページを参照してください。

**参照**

- 「SET OPTION 文」『SQL Anywhere サーバー SQL リファレンス』
- 「データベースオプション」568 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』

## SQL Anywhere SNMP Extension Agent によるストアドプロシージャーの実行

SQL Anywhere MIB には、SQL Anywhere SNMP Extension Agent を使用してストアドプロシージャーを実行するための OID が格納されています。ストアドプロシージャーを実行するには、SQL Anywhere SNMP Extension Agent が接続に使用するユーザーが以下に示す要件のいずれか 1 つを満たしている必要があります。

●プロシージャーの実行パーミッションを持っている

●プロシージャーの所有者である

●DBA 権限を持っている

プロシージャーによって生成された結果セットや戻り値は無視されます。

SQL Anywhere SNMP Extension Agent を使用してストアドプロシージャーを実行するには、saAgent.saProc (OID 1.3.6.1.4.1.897.2.3.5.*db*、*db* は *sasnmp.ini* ファイル内のデータベース番号) の値としてストアドプロシージャーの名前を示す文字列を設定します。必要に応じて、プロシージャーに引数を指定できます。引数を指定しなかった場合には、プロシージャー名の後に空のカッコが追加されます。

たとえば、saAgent.saProc の値として文字列 "pchin.updatesales( 'param1', 2)" を設定すると、ユーザー pchin が所有するストアドプロシージャー updatesales が呼び出されます。

この OID の値としてプロシージャー名を設定する方法は、使用する SNMP 管理ソフトウェアによって異なります。 「SQL Anywhere MIB」1321 ページを参照してください。

## トラップ

「トラップ」は、特定のイベントが発生した場合に SNMP エージェントから送信される OID です。トラップは SNMP エージェントで開始され、SNMP 管理ソフトウェアによって検出できます。検出後、SNMP 管理ソフトウェアは、イベントを直接処理することも、SNMP エージェントに詳細を問い合わせることもできます。

トラップを受信するには、SNMP サービスを設定する必要があります。SNMP サービスはトラップ情報を受信してこれを転送しますが、デフォルトではどこにも転送されず、動作中のトラップリスナーは何も検出しません。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

♦ **SNMP サービスを設定してコンピューターにトラップを送信するには、次の手順に従います。**

1. **[マイ コンピューター]** を右クリックし、**[管理]** をクリックします。
2. 左ウィンドウ枠で、**[サービスとアプリケーション]** をダブルクリックします。
3. 左ウィンドウ枠で、**[サービス]** をダブルクリックします。
4. 右ウィンドウ枠のサービスのリストで [SNMP Service] を見つけ、右クリックして、**[プロパティ]** をクリックします。
5. **[トラップ]** タブをクリックします。
6. **[追加]** をクリックします。
7. **[SNMP サービスの構成]** ウィンドウで、テキストボックスに **localhost** と入力し、**[追加]** をクリックします。
8. **[OK]** をクリックします。

### SQL Anywhere SNMP Extension Agent トラップ

SQL Anywhere SNMP Extension Agent は、データベースサーバーによって接続が停止されると、必ずトラップを送信します。このトラップの OID は 1.3.6.1.2.1.39.2.1 です。

データベースミラーリングを使用している場合に、SQL Anywhere SNMP Extension Agent とデータベースサーバーとの接続が切断されると、SQL Anywhere SNMP Extension Agent は同じデータベースサーバーへの再接続を 30 秒おきに試行します。再接続が成功しても、接続先が以前とは別のデータベースサーバーであった場合 (ServerName プロパティによって確認できます)、SQL Anywhere SNMP Extension Agent は OID 1.3.6.1.4.1.897.2.6.3 のトラップと *sasnmp.ini* ファイルから取得したデータベース ID を送信します。この場合、SQL Anywhere SNMP Extension Agent が接続していたプライマリデータベースサーバーが停止したため、現在はミラーサーバーがプライマリサーバーとして動作しています。「データベースミラーリング」1137 ページを参照してください。

SQL Anywhere SNMP Extension Agent が送信する他のトラップは、すべて動的トラップです。「動的トラップ」1331 ページを参照してください。

## 動的トラップ

「動的トラップ」は、特定のプロパティ、統計、またはオプションの値を含む単純な式の評価が true である場合に SQL Anywhere Extension Agent によって送信されるトラップです。動的トラップは、*sasnmp.ini* ファイル内に作成されます。*sasnmp.ini* ファイルエントリ内のトラップ情報のフォーマットは次のとおりです。

```
Traptrapnum=[1.3.6.1.4.1.897.2.]oid[.dbnum] op value
```

***trapnum*** 動的トラップの番号です。開始値が 1 の連番になっている必要があります。

***oid*** プロパティ、統計、またはオプションの OID です。SQL Anywhere MIB 内または RDBMS MIB 内の OID がサポートされます。指定された OID が有効な SQL Anywhere OID または RDBMS OID ではない場合は、SQL Anywhere MIB プレフィクス 1.3.6.1.4.1.897.2. が先頭に追加されます。

SQL Anywhere MIB 内の OID の詳細については、「SQL Anywhere MIB リファレンス」1333 ページを参照してください。

RDBMS MIB 内の OID の詳細については、「RDBMS MIB リファレンス」1372 ページを参照してください。

**注意**
動的トラップでは、データベースサーバーまたはデータベースのプロパティ、統計、またはオプションに対応する OID のみ使用できます。

***dbnum*** データベース番号です。このフィールドはオプションですが、指定する場合は *sasnmp.ini* ファイルの [DB*n*] セクションのデータベース番号と一致する番号を指定する必要があります。

次のいずれかの値を指定します。

○= または == (等しい)

○**!=**、<>、または >< (等しくない)

○<= または =< (以下)

○>= または => (以上)

○< (より小さい)

○> (より大きい)

***op***

**注意**
値が文字列の場合、等しいか等しくないかの評価のみがサポートされます。

***値*** 式内で使用する値です。文字列値は、一重引用符または二重引用符で囲む必要があります。これらの引用符は値の一部として扱われません。開始引用符または終了引用符を文字列に含める必要がある場合は、それらを二重に指定する必要があります。文字列内に出現する一重引用符は二重引用符にしません。

動的トラップを設定するとき、単位をキロバイト、メガバイト、ギガバイト、またはテラバイトで指定するには、それぞれ k、m、g、または t を使用します。たとえば、次のように指定することで、現在のキャッシュサイズが 200 MB を超えた場合にトラップがトリガーするように、動的トラップを設定できます。

```
Trap1=1.3.6.1.4.1.897.2.1.1.11.1 > 200M
```

*sasnmp.ini* ファイル内には、任意の数の Trap フィールドを指定できます。トラップに使用される OID は 1.3.6.1.4.1.897.2.4.1 です。トラップとともに送信されるデータには次のものがあります。

○トラップ番号 (SQL Anywhere SNMP Agent によって送信される動的トラップの番号は開始値が 1 の連番)

○データベースインデックス

○データベース名トラップインデックス (*sasnmp.ini* ファイル内の値)

○変数名

○変数値 (変数の現在値。閾値になるとはかぎらない。)

### 動的トラップの動作

動的トラップは、いったんトリガーされると、トリガーを引き起こした条件が FALSE に変わり、その後再び TRUE になるまで、再送信されません。

たとえば、1.1.11.1 >= 51200K という式を使用して動的トラップを設定した場合、サーバーのキャッシュサイズが 50 MB (51200 KB) に達すると、トラップがトリガーされます。この時点で、この動的トラップは無効になり、それ以上送信されません。このトラップが再び有効になるのは、キャッシュサイズがその後 50 MB 未満になった場合だけです。この後でキャッシュサイズが 50 MB に戻ると、そのことが通知されます。

### トラップの例

| トラップ情報 | 説明 |
|---|---|
| Trap1=1.1.5 > 10000 | サーバーから送信されたデータが 10000 バイトを超えると、トラップが送信される。 |
| Trap2=1.3.6.1.2.1.39.1.4.1.4.14.1 = 10485760 | トランザクションログファイルのサイズが 10 MB を超えると、トラップが送信される。 |

## SQL Anywhere MIB リファレンス

SNMP エージェントがサポートするオブジェクト識別子 (OID) のリストは、「Management Information Base (MIB)」と呼ばれるファイルに保存されています。このリストには、OID の名前や型などの情報も含まれています。以下の項では、SQL Anywhere SNMP Extension Agent を使用して検索や設定ができる統計、プロパティ、オプションの OID を示します。

### 参照

● 「SNMP」1320 ページ

## Agent

Agent テーブルには、SQL Anywhere SNMP Extension Agent の情報が含まれています。

書き込み可能なプロパティには、アスタリスク記号 (*) が付いています。値 *n* は、*sasnmp.ini* ファイル内のデータベース番号です。

| OID | 型 | 名前 | 戻り値 |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.3.1 | String | saVersion | エージェントのバージョン |
| 1.3.6.1.4.1.897.2.3.2.*n* | String | saDBConnStr | 接続文字列 |
| 1.3.6.1.4.1.897.2.3.3.*n* | Integer32 | saConnected* | エージェントが接続されている場合は 1。それ以外の場合は 0。 |
| 1.3.6.1.4.1.897.2.3.4.*n* | Integer32 | saStarted* | データベースが起動されている場合は 1。それ以外の場合は 0。 |
| 1.3.6.1.4.1.897.2.3.5.*n* | String | saProc* | " " |
| 1.3.6.1.4.1.897.2.3.6 | String | saRestart* | 0 |

## asaMetaData テーブル

SQL Anywhere MIB には、次のメタデータテーブルがあります。

- saSrvMetaData.saSrvStatMetaDataTable
- saSrvMetaData.saSrvPropMetaDataTable
- saSrvMetaData.saDbStatMetaDataTable
- saSrvMetaData.saDbPropMetaDataTable
- saSrvMetaData.saDbOptMetaDataTable

### saSrvMetaData.saSrvStatMetaDataTable

このテーブルには、データベースサーバー統計についてのメタデータが格納されています。

値 *db* は、*sasnmp.ini* ファイル内のデータベース番号です。

| OID | 型 | 名前 | 戻り値 |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.4.1.1.1.1.*db* | Integer32 | saSrvStatIndex | *db* |
| 1.3.6.1.4.1.897.2.4.1.1.1.2.*db* | Integer32 | saSrvStatObjType | 1[1] |
| 1.3.6.1.4.1.897.2.4.1.1.1.3.*db* | Integer32 | saSrvStatType | 1[2] |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| OID | 型 | 名前 | 戻り値 |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.4.1.1.1.4.*db* | OID | saSrvStatOID | SQL Anywhere MIB エントリの OID[3] |
| 1.3.6.1.4.1.897.2.4.1.1.1.5.*db* | String | saSrvStatName | 統計名 |

[1] 値：1 = サーバー、2 = データベース

[2] 値：1 = 統計、2 = プロパティ、3 = オプション

[3] 返される OID にデータベース番号は含まれません。クエリ内でデータベース番号を使用するためには、あらかじめ OID にそのデータベース番号を追加しておく必要があります。

## saSrvMetaData.saSrvPropMetaDataTable

このテーブルには、データベースサーバープロパティについてのメタデータが格納されています。

値 *db* は、*sasnmp.ini* ファイル内のデータベース番号です。

| OID | 型 | 名前 | 戻り値 |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.4.1.2.1.1.*db* | Integer32 | saSrvPropIndex | *db* |
| 1.3.6.1.4.1.897.2.4.1.2.1.2.*db* | Integer32 | saSrvPropObjType | 1[1] |
| 1.3.6.1.4.1.897.2.4.1.2.1.3.*db* | Integer32 | saSrvPropType | 2[2] |
| 1.3.6.1.4.1.897.2.4.1.2.1.4.*db* | OID | saSrvPropOID | SQL Anywhere MIB エントリの OID[3] |
| 1.3.6.1.4.1.897.2.4.1.2.1.5.*db* | String | saSrvPropName | プロパティ名 |

[1] 値：1 = サーバー、2 = データベース

[2] 値：1 = 統計、2 = プロパティ、3 = オプション

[3] 返される OID にデータベース番号は含まれません。クエリ内でデータベース番号を使用するためには、あらかじめ OID にそのデータベース番号を追加しておく必要があります。

## saDbMetaData.saDbStatMetaDataTable

このテーブルには、データベース統計についてのメタデータが格納されています。

値 *db* は、*sasnmp.ini* ファイル内のデータベース番号です。

| OID | 型 | 名前 | 戻り値 |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.4.2.1.1.1.*db* | Integer32 | saDbStatIndex | *db* |
| 1.3.6.1.4.1.897.2.4.2.1.1.2.*db* | Integer32 | saDbStatObjType | 2[1] |
| 1.3.6.1.4.1.897.2.4.2.1.1.3.*db* | Integer32 | saDbStatType | 1[2] |
| 1.3.6.1.4.1.897.2.4.2.1.1.4.*db* | OID | saDbStatOID | SQL Anywhere MIB エントリの OID[3] |
| 1.3.6.1.4.1.897.2.4.2.1.1.5.*db* | String | saDbStatName | 統計名 |

[1] 値：1 = サーバー、2 = データベース

[2] 値：1 = 統計、2 = プロパティ、3 = オプション

[3] 返される OID にデータベース番号は含まれません。クエリ内でデータベース番号を使用するためには、あらかじめ OID にそのデータベース番号を追加しておく必要があります。

## saDbMetaData.saDbPropMetaDataTable

このテーブルには、データベースプロパティについてのメタデータが格納されています。

値 *db* は、*sasnmp.ini* ファイル内のデータベース番号です。

| OID | 型 | 名前 | 戻り値 |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.4.2.2.1.1.*db* | Integer32 | saDbPropIndex | *db* |
| 1.3.6.1.4.1.897.2.4.2.2.1.2.*db* | Integer32 | saDbPropObjType | 2[1] |
| 1.3.6.1.4.1.897.2.4.2.2.1.3.*db* | Integer32 | saDbPropType | 2[2] |
| 1.3.6.1.4.1.897.2.4.2.2.1.4.*db* | OID | saDbPropOID | SQL Anywhere MIB エントリの OID[3] |
| 1.3.6.1.4.1.897.2.4.2.2.1.5.*db* | String | saDbPropName | プロパティ名 |

[1] 値：1 = サーバー、2 = データベース

[2] 値：1 = 統計、2 = プロパティ、3 = オプション

[3] 返される OID にデータベース番号は含まれません。クエリ内でデータベース番号を使用するためには、あらかじめ OID にそのデータベース番号を追加しておく必要があります。

### saDbMetaData.saDbOptMetaDataTable

このテーブルには、データベースオプションについてのメタデータが格納されています。

値 *db* は、*sasnmp.ini* ファイル内のデータベース番号です。

| OID | 型 | 名前 | 戻り値 |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.4.2.1.1.1.*db* | Integer32 | saDbOptIndex | *db* |
| 1.3.6.1.4.1.897.2.4.2.1.1.2.*db* | Integer32 | saDbOptObjType | 2[1] |
| 1.3.6.1.4.1.897.2.4.2.1.1.3.*db* | Integer32 | saDbOptType | 3[2] |
| 1.3.6.1.4.1.897.2.4.2.1.1.4.*db* | OID | saDbOptOID | SQL Anywhere MIB エントリの OID[3] |
| 1.3.6.1.4.1.897.2.4.2.1.1.5.*db* | String | saDbOptName | オプション名 |

[1] 値：1 = サーバー、2 = データベース

[2] 値：1 = 統計、2 = プロパティ、3 = オプション

[3] 返される OID にデータベース番号は含まれません。クエリ内でデータベース番号を使用するためには、あらかじめ OID にそのデータベース番号を追加しておく必要があります。

## SQL Anywhere MIB サーバー統計

次の表は、SQL Anywhere SNMP Extension Agent を使用して検索できるデータベースサーバー統計の OID と名前を示します。

値 *n* は、*sasnmp.ini* ファイル内のデータベース番号です。

データベースサーバー統計の詳細については、「データベースサーバープロパティ値のアクセス」753 ページと「データベースプロパティ値のアクセス」781 ページを参照してください。

### 説明

| OID | 型 | 名前 | 統計情報 |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatActiveReq | ActiveReq |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatAvailIO | AvailIO |
| 1.3.6.1.4.1.897.2.1.1.*n* | Counter64 | srvStatBytesReceived | BytesReceived |

| OID | 型 | 名前 | 統計情報 |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.1.1.*n* | Counter64 | srvStatBytesReceivedUncomp | BytesReceivedUncomp |
| 1.3.6.1.4.1.897.2.1.1.*n* | Counter64 | srvStatBytesSent | BytesSent |
| 1.3.6.1.4.1.897.2.1.1.*n* | Counter64 | srvStatBytesSentUncomp | BytesSentUncomp |
| 1.3.6.1.4.1.897.2.1.1.*n* | Counter64 | srvStatCacheHits | CacheHits |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatCachePinned | CachePinned |
| 1.3.6.1.4.1.897.2.1.1.*n* | Counter64 | srvStatCacheRead | CacheRead |
| 1.3.6.1.4.1.897.2.1.1.*n* | Counter64 | srvStatCacheReplacements | CacheReplacements |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatCurrentCacheSize | CurrentCacheSize |
| 1.3.6.1.4.1.897.2.1.1.*n* | Counter64 | srvStatDiskRead | DiskRead |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatFreeBuffers | FreeBuffers |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatInternal | Internal |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatUniqueClientAddresses | UniqueClientAddresses |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatLockedHeapPages | LockedHeapPages |
| 1.3.6.1.4.1.897.2.1.1.*n* | Counter64 | srvStatMainHeapBytes | MainHeapBytes |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatMainHeapPages | MainHeapPages |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatMapPhysicalMemoryEng | MapPhysicalMemoryEng |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatMaxCacheSize | MaxCacheSize |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatMinCacheSize | MinCacheSize |
| 1.3.6.1.4.1.897.2.1.1.*n* | Counter64 | srvStatMultiPacketsReceived | MultiPacketsReceived |

| OID | 型 | 名前 | 統計情報 |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.1.1.*n* | Counter64 | srvStatMultiPacketsSent | MultiPacketsSent |
| 1.3.6.1.4.1.897.2.1.1.*n* | Counter64 | srvStatPacketsReceived | PacketsReceived |
| 1.3.6.1.4.1.897.2.1.1.*n* | Counter64 | srvStatPacketsReceivedUncomp | PacketsReceivedUncomp |
| 1.3.6.1.4.1.897.2.1.1.*n* | Counter64 | srvStatPacketsSent | PacketsSent |
| 1.3.6.1.4.1.897.2.1.1.*n* | Counter64 | srvStatPacketsSentUncomp | PacketsSentUncomp |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatPeakCacheSize | PeakCacheSize |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatRemoteputWait | RemoteputWait |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatReq | Req |
| 1.3.6.1.4.1.897.2.1.1.*n* | Counter64 | srvStatSendFail | SendFail |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatTotalBuffers | TotalBuffers |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatUnschReq | UnschReq |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatInternal | Internal |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatCacheFile | CacheFile |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatCacheFileDirty | CacheFileDirty |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatCacheAllocated | CacheAllocated |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatCachePanics | CachePanics |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatCacheFree | CacheFree |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatCacheScavenges | CacheScavenges |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatCacheScavengeVisited | CacheScavengeVisited |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatLockedCursorPages | LockedCursorPages |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatQueryHeapPages | QueryHeapPages |

| OID | 型 | 名前 | 統計情報 |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatCarverHeapPages | CarverHeapPages |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatHeapsRelocatable | HeapsRelocatable |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatHeapsLocked | HeapsLocked |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatHeapsQuery | HeapsQuery |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatHeapsCarver | HeapsCarver |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatMultiPageAllocs | MultiPageAllocs |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatRequestsReceived | RequestsReceived |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatExchangeTasks | ExchangeTasks |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatClientStmtCacheHits | ClientStmtCacheHits |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatClientStmtCacheMisses | ClientStmtCacheMisses |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatQueryMemActiveCurr | QueryMemActiveCurr |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatQueryMemActiveEst | QueryMemActiveEst |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatQueryMemGrantWaiting | QueryMemGrantWaiting |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatQueryMemGrantRequested | QueryMemGrantRequested |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatQueryMemGrantWaited | QueryMemGrantWaited |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatQueryMemGrantFailed | QueryMemGrantFailed |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatQueryMemGrantGranted | QueryMemGrantGranted |

| OID | 型 | 名前 | 統計情報 |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatQueryMemExtraAvail | QueryMemExtraAvail |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatThreadDeadlocksAvoided | ThreadDeadlocksAvoided |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatThreadDeadlocksReported | ThreadDeadlocksReported |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatCommit | Commit |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatCursor | Cursor |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatCursorOpen | CursorOpen |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatRlbk | Rlbk |
| 1.3.6.1.4.1.897.2.1.1.*n* | Integer32 | srvStatPrepStmt | PrepStmt |

## SQL Anywhere MIB サーバープロパティ

次の表は、SQL Anywhere SNMP Extension Agent を使用して検索できるデータベースサーバープロパティの OID と名前を示します。

書き込み可能なプロパティには、アスタリスク記号 (*) が付いています。値 *n* は、*sasnmp.ini* ファイル内のデータベース番号です。

### 説明

| OID | 型 | 名前 | プロパティ |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.1.2.1.*n* | String | srvPropInternal | Internal |
| 1.3.6.1.4.1.897.2.1.2.2.*n* | String | srvPropCharSet | CharSet |
| 1.3.6.1.4.1.897.2.1.2.3.*n* | String | srvPropCommandLine | CommandLine |
| 1.3.6.1.4.1.897.2.1.2.4.*n* | String | srvPropCompactPlatformVer | CompactPlatformVer |

| OID | 型 | 名前 | プロパティ |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.1.2.5.*n* | String | srvPropCompanyName | CompanyName |
| 1.3.6.1.4.1.897.2.1.2.6.*n* | String | srvPropConnsDisabled* | ConnsDisabled |
| 1.3.6.1.4.1.897.2.1.2.7.*n* | String | srvPropConsoleLogFile | ConsoleLogFile |
| 1.3.6.1.4.1.897.2.1.2.8.*n* | String | srvPropDefaultCollation | DefaultCollation |
| 1.3.6.1.4.1.897.2.1.2.9.*n* | String | srvPropIdleTimeout | IdleTimeout |
| 1.3.6.1.4.1.897.2.1.2.10.*n* | String | srvPropIsIQ | IsIQ |
| 1.3.6.1.4.1.897.2.1.2.11.*n* | String | srvPropInternal | Internal |
| 1.3.6.1.4.1.897.2.1.2.12.*n* | String | srvPropIsNetworkServer | IsNetworkServer |
| 1.3.6.1.4.1.897.2.1.2.13.*n* | String | srvPropIsRuntimeServer | IsRuntimeServer |
| 1.3.6.1.4.1.897.2.1.2.14.*n* | String | srvPropInternal | Internal |
| 1.3.6.1.4.1.897.2.1.2.15.*n* | String | srvPropLanguage | Language |
| 1.3.6.1.4.1.897.2.1.2.16.*n* | String | srvPropLegalCopyright | LegalCopyright |
| 1.3.6.1.4.1.897.2.1.2.17.*n* | String | srvPropLegalTrademarks | LegalTrademarks |
| 1.3.6.1.4.1.897.2.1.2.18.*n* | String | srvPropLicenseCount | LicenseCount |
| 1.3.6.1.4.1.897.2.1.2.19.*n* | String | srvPropLicensedCompany | LicensedCompany |
| 1.3.6.1.4.1.897.2.1.2.20.*n* | String | srvPropLicensedUser | LicensedUser |

| OID | 型 | 名前 | プロパティ |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.1.2.21.*n* | String | srvPropLicenseType | LicenseType |
| 1.3.6.1.4.1.897.2.1.2.22.*n* | String | srvPropLivenessTimeout* | LivenessTimeout |
| 1.3.6.1.4.1.897.2.1.2.23.*n* | String | srvPropMachineName | MachineName |
| 1.3.6.1.4.1.897.2.1.2.24.*n* | String | srvPropMaxMessage | MaxMessage |
| 1.3.6.1.4.1.897.2.1.2.25.*n* | String | srvPropMessageWindowSize | MessageWindowSize |
| 1.3.6.1.4.1.897.2.1.2.26.*n* | String | srvPropName | Name |
| 1.3.6.1.4.1.897.2.1.2.27.*n* | String | srvPropNativeProcessorArchitecture | NativeProcessorArchitecture |
| 1.3.6.1.4.1.897.2.1.2.28.*n* | String | srvPropNumPhysicalProcessors | NumPhysicalProcessors |
| 1.3.6.1.4.1.897.2.1.2.29.*n* | String | srvPropInternal | Internal |
| 1.3.6.1.4.1.897.2.1.2.30.*n* | String | srvPropOmniIdentifier | OmniIdentifier |
| 1.3.6.1.4.1.897.2.1.2.31.*n* | String | srvPropPageSize | PageSize |
| 1.3.6.1.4.1.897.2.1.2.32.*n* | String | srvPropPlatform | Platform |
| 1.3.6.1.4.1.897.2.1.2.33.*n* | String | srvPropPlatformVer | PlatformVer |
| 1.3.6.1.4.1.897.2.1.2.34.*n* | String | srvPropProcessCPU | ProcessCPU |
| 1.3.6.1.4.1.897.2.1.2.35.*n* | String | srvPropProcessCPUSystem | ProcessCPUSystem |
| 1.3.6.1.4.1.897.2.1.2.36.*n* | String | srvPropProcessCPUUser | ProcessCPUUser |

| OID | 型 | 名前 | プロパティ |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.1.2.37.*n* | String | srvPropProcessorArchitecture | ProcessorArchitecture |
| 1.3.6.1.4.1.897.2.1.2.38.*n* | String | srvPropProductName | ProductName |
| 1.3.6.1.4.1.897.2.1.2.39.*n* | String | srvPropProductVersion | ProductVersion |
| 1.3.6.1.4.1.897.2.1.2.40.*n* | String | srvPropQuittingTime* | QuittingTime |
| 1.3.6.1.4.1.897.2.1.2.41.*n* | String | srvPropRememberLastStatement* | RememberLastStatement |
| 1.3.6.1.4.1.897.2.1.2.42.*n* | String | srvPropRequestFilterConn | RequestFilterConn |
| 1.3.6.1.4.1.897.2.1.2.43.*n* | String | srvPropRequestFilterDB | RequestFilterDB |
| 1.3.6.1.4.1.897.2.1.2.44.*n* | String | srvPropRequestLogFile* | RequestLogFile |
| 1.3.6.1.4.1.897.2.1.2.45.*n* | String | srvPropRequestLogging* | RequestLogging |
| 1.3.6.1.4.1.897.2.1.2.46.*n* | String | srvPropRequestLogMaxSize | RequestLogMaxSize |
| 1.3.6.1.4.1.897.2.1.2.47.*n* | String | srvPropStartTime | StartTime |
| 1.3.6.1.4.1.897.2.1.2.48.*n* | String | srvPropTempDir | TempDir |
| 1.3.6.1.4.1.897.2.1.2.49.*n* | String | srvPropMultiProgrammingLevel | MultiProgrammingLevel |
| 1.3.6.1.4.1.897.2.1.2.50.*n* | String | srvPropTimeZoneAdjustment | TimeZoneAdjustment |
| 1.3.6.1.4.1.897.2.1.2.51.*n* | String | srvPropHttpPorts | HttpPorts |
| 1.3.6.1.4.1.897.2.1.2.52.*n* | String | srvPropHttpsPorts | HttpsPorts |

| OID | 型 | 名前 | プロパティ |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.1.2.53.*n* | String | srvPropProfileFilterConn | ProfileFilterConn |
| 1.3.6.1.4.1.897.2.1.2.54.*n* | String | srvPropProfileFilterUser | ProfileFilterUser |
| 1.3.6.1.4.1.897.2.1.2.55.*n* | String | srvPropRequestLogNumFiles | RequestLogNumFiles |
| 1.3.6.1.4.1.897.2.1.2.56.*n* | String | srvPropIsFipsAvailable | IsFipsAvailable |
| 1.3.6.1.4.1.897.2.1.2.57.*n* | String | srvPropFipsMode | FipsMode |
| 1.3.6.1.4.1.897.2.1.2.58.*n* | String | srvPropStartDBPermission | StartDBPermission |
| 1.3.6.1.4.1.897.2.1.2.59.*n* | String | srvPropServerName | ServerName |
| 1.3.6.1.4.1.897.2.1.2.60.*n* | String | srvPropRememberLastPlan | RememberLastPlan |
| 1.3.6.1.4.1.897.2.1.2.61.*n* | String | srvPropInternal | Internal |
| 1.3.6.1.4.1.897.2.1.2.62.*n* | String | srvPropInternal | Internal |
| 1.3.6.1.4.1.897.2.1.2.63.*n* | String | srvPropRequestTiming | RequestTiming |
| 1.3.6.1.4.1.897.2.1.2.64.*n* | String | srvPropCacheSizingStatistics | CacheSizingStatistics |
| 1.3.6.1.4.1.897.2.1.2.65.*n* | String | srvPropConsoleLogMaxSize | ConsoleLogMaxSize |
| 1.3.6.1.4.1.897.2.1.2.66.*n* | String | srvPropDebuggingInformation | DebuggingInformation |
| 1.3.6.1.4.1.897.2.1.2.67.*n* | String | srvPropMessage | Message |
| 1.3.6.1.4.1.897.2.1.2.68.*n* | String | srvPropMessageText | MessageText |

| OID | 型 | 名前 | プロパティ |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.1.2.69.*n* | String | srvPropMessageTime | MessageTime |
| 1.3.6.1.4.1.897.2.1.2.70.*n* | String | srvPropIsRsaAvailable | IsRsaAvailable |
| 1.3.6.1.4.1.897.2.1.2.71.*n* | String | srvPropIsEccAvailable | IsEccAvailable |
| 1.3.6.1.4.1.897.2.1.2.72.*n* | String | srvPropMaxConnections | MaxConnections |
| 1.3.6.1.4.1.897.2.1.2.73.*n* | String | srvPropNumLogicalProcessors | NumLogicalProcessors |
| 1.3.6.1.4.1.897.2.1.2.74.*n* | String | srvPropNumLogicalProcessorsUsed | NumLogicalProcessorsUsed |
| 1.3.6.1.4.1.897.2.1.2.75.*n* | String | srvPropNumPhysicalProcessorsUsed | NumPhysicalProcessorsUsed |
| 1.3.6.1.4.1.897.2.1.2.76.*n* | String | srvPropDefaultNcharCollation | DefaultNcharCollation |
| 1.3.6.1.4.1.897.2.1.2.77.*n* | String | srvPropCollectStatistics | CollectStatistics |
| 1.3.6.1.4.1.897.2.1.2.78.*n* | String | srvPropFirstOption | FirstOption |
| 1.3.6.1.4.1.897.2.1.2.79.*n* | String | srvPropLastOption | LastOption |
| 1.3.6.1.4.1.897.2.1.2.80.*n* | String | srvPropLastConnectionProperty | LastConnectionProperty |
| 1.3.6.1.4.1.897.2.1.2.81.*n* | String | srvPropLastDatabaseProperty | LastDatabaseProperty |
| 1.3.6.1.4.1.897.2.1.2.82.*n* | String | srvPropLastServerProperty | LastServerProperty |
| 1.3.6.1.4.1.897.2.1.2.83.*n* | String | srvPropWebClientLogging | WebClientLogging |
| 1.3.6.1.4.1.897.2.1.2.84.*n* | String | srvPropWebClientLogFile | WebClientLogFile |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| OID | 型 | 名前 | プロパティ |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.1.2.85.*n* | String | srvPropHttpNumConnections | HttpNumConnections |
| 1.3.6.1.4.1.897.2.1.2.86.*n* | String | srvPropHttpsNumConnections | HttpsNumConnections |
| 1.3.6.1.4.1.897.2.1.2.87.*n* | String | srvPropInternal | Internal |
| 1.3.6.1.4.1.897.2.1.2.88.*n* | String | srvPropHttpNumActiveReq | HttpNumActiveReq |
| 1.3.6.1.4.1.897.2.1.2.89.*n* | String | srvPropHttpsNumActiveReq | HttpsNumActiveReq |
| 1.3.6.1.4.1.897.2.1.2.90.*n* | String | srvPropHttpNumSessions | HttpNumSessions |
| 1.3.6.1.4.1.897.2.1.2.91.*n* | String | srvPropQueryMemPages | QueryMemPages |
| 1.3.6.1.4.1.897.2.1.2.92.*n* | String | srvPropQueryMemGrantBase | QueryMemGrantBase |
| 1.3.6.1.4.1.897.2.1.2.93.*n* | String | srvPropQueryMemGrantBaseMI | QueryMemGrantBaseMI |
| 1.3.6.1.4.1.897.2.1.2.94.*n* | String | srvPropQueryMemGrantExtra | QueryMemGrantExtra |
| 1.3.6.1.4.1.897.2.1.2.95.*n* | String | srvPropQueryMemActiveMax | QueryMemActiveMax |
| 1.3.6.1.4.1.897.2.1.2.96.*n* | String | srvPropQueryMemPercentOfCache | QueryMemPercentOfCache |
| 1.3.6.1.4.1.897.2.1.2.97.*n* | String | srvPropMessageCategoryLimit | MessageCategoryLimit |
| 1.3.6.1.4.1.897.2.1.2.98.*n* | String | srvPropInternal | Internal |
| 1.3.6.1.4.1.897.2.1.2.99.*n* | String | srvPropInternal | Internal |
| 1.3.6.1.4.1.897.2.1.2.100.*n* | String | srvPropIsService | IsService |

| OID | 型 | 名前 | プロパティ |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.1.2.101.*n* | String | srvPropTcpIpAddresses | TcpIpAddresses |
| 1.3.6.1.4.1.897.2.1.2.102.*n* | String | srvPropHttpAddresses | HttpAddresses |
| 1.3.6.1.4.1.897.2.1.2.103.*n* | String | srvPropHttpsAddresses | HttpsAddresses |
| 1.3.6.1.4.1.897.2.1.2.104.*n* | String | srvPropProcessID | ProcessID |
| 1.3.6.1.4.1.897.2.1.2.105.*n* | String | srvPropRemoteCapability | RemoteCapability |
| 1.3.6.1.4.1.897.2.1.2.106.*n* | String | srvPropMaxRemoteCapability | MaxRemoteCapability |
| 1.3.6.1.4.1.897.2.1.2.107.*n* | String | srvPropEventTypeName | EventTypeName |
| 1.3.6.1.4.1.897.2.1.2.108.*n* | String | srvPropEventTypeDesc | EventTypeDesc |
| 1.3.6.1.4.1.897.2.1.2.109.*n* | String | srvPropMaxEventType | MaxEventType |
| 1.3.6.1.4.1.897.2.1.2.110.*n* | String | srvPropIsPortableDevice | IsPortableDevice |
| 1.3.6.1.4.1.897.2.1.2.111.*n* | String | srvPropIPAddressMonitorPeriod | IPAddressMonitorPeriod |
| 1.3.6.1.4.1.897.2.1.2.112.*n* | String | srvPropServerEdition | ServerEdition |
| 1.3.6.1.4.1.897.2.1.2.113.*n* | String | srvPropObjectType | ObjectType |
| 1.3.6.1.4.1.897.2.1.2.114.*n* | String | srvPropCurrentMultiProgrammingLevel | CurrentMultiProgrammingLevel |
| 1.3.6.1.4.1.897.2.1.2.115.*n* | String | srvPropMinMultiProgrammingLevel | MinMultiProgrammingLevel |
| 1.3.6.1.4.1.897.2.1.2.116.*n* | String | srvPropMaxMultiProgrammingLevel | MaxMultiProgrammingLevel |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| OID | 型 | 名前 | プロパティ |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.1.2.117.*n* | String | srvPropAutoMultiProgrammingLevel | AutoMultiProgrammingLevel |
| 1.3.6.1.4.1.897.2.1.2.118.*n* | String | srvPropAutoMultiProgrammingLevelStatistics | AutoMultiProgrammingLevelStatistics |
| 1.3.6.1.4.1.897.2.1.2.119.*n* | String | srvPropApproximateCPUTime | ApproximateCPUTime |
| 1.3.6.1.4.1.897.2.1.2.120.*n* | String | srvPropConnectedTime | ConnectedTime |
| 1.3.6.1.4.1.897.2.1.2.121.*n* | String | srvPropReqCountUnscheduled | ReqCountUnscheduled |
| 1.3.6.1.4.1.897.2.1.2.122.*n* | String | srvPropReqCountActive | ReqCountActive |
| 1.3.6.1.4.1.897.2.1.2.123.*n* | String | srvPropReqCountBlockIO | ReqCountBlockIO |
| 1.3.6.1.4.1.897.2.1.2.124.*n* | String | srvPropReqCountBlockLock | ReqCountBlockLock |
| 1.3.6.1.4.1.897.2.1.2.125.*n* | String | srvPropReqCountBlockContention | ReqCountBlockContention |
| 1.3.6.1.4.1.897.2.1.2.126.*n* | String | srvPropReqTimeUnscheduled | ReqTimeUnscheduled |
| 1.3.6.1.4.1.897.2.1.2.127.*n* | String | srvPropReqTimeActive | ReqTimeActive |
| 1.3.6.1.4.1.897.2.1.2.128.*n* | String | srvPropReqTimeBlockIO | ReqTimeBlockIO |
| 1.3.6.1.4.1.897.2.1.2.129.*n* | String | srvPropReqTimeBlockLock | ReqTimeBlockLock |
| 1.3.6.1.4.1.897.2.1.2.130.*n* | String | srvPropReqTimeBlockContention | ReqTimeBlockContention |
| 1.3.6.1.4.1.897.2.1.2.131.*n* | String | srvPropLicenseKey | LicenseKey |

### 参照

- 「データベースサーバープロパティ値のアクセス」753 ページ

## SQL Anywhere MIB データベース統計

次の表は、SQL Anywhere SNMP Extension Agent を使用して検索できるデータベース統計の OID と名前を示します。

値 *n* は、*sasnmp.ini* ファイル内のデータベース番号です。

### 説明

| OID | 型 | 名前 | 統計情報 |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.1.*n* | Counter64 | dbStatCacheHits | CacheHits |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatCacheReadIndInt | CacheReadIndInt |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatCacheReadIndLeaf | CacheReadIndLeaf |
| 1.3.6.1.4.1.897.2.2.1.*n* | Counter64 | dbStatCacheRead | CacheRead |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatCacheReadTable | CacheReadTable |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatChkpt | Chkpt |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatChkptFlush | ChkptFlush |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatChkptPage | ChkptPage |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatCheckpointUrgency | CheckpointUrgency |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatInternal | Internal |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatInternal | Internal |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatCheckpointLogCommitToDisk | CheckpointLogCommitToDisk |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatCheckpointLogPagesInUse | CheckpointLogPagesInUse |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatCheckpointLogPagesRelocated | CheckpointLogPagesRelocated |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| OID | 型 | 名前 | 統計情報 |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatCheckpointLogSavePreimage | CheckpointLogSavePreimage |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatCheckpointLogSize | CheckpointLogSize |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatCheckpointLogWrites | CheckpointLogWrites |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatCheckpointLogPagesWritten | CheckpointLogPagesWritten |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatCommitFile | CommitFile |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatConnCount | ConnCount |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatCurrIO | CurrIO |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatCurrRead | CurrRead |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatCurrWrite | CurrWrite |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatDiskReadIndInt | DiskReadIndInt |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatDiskReadIndLeaf | DiskReadIndLeaf |
| 1.3.6.1.4.1.897.2.2.1.*n* | Counter64 | dbStatDiskRead | DiskRead |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatDiskReadTable | DiskReadTable |
| 1.3.6.1.4.1.897.2.2.1.*n* | Counter64 | dbStatDiskWrite | DiskWrite |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatExtendDB | ExtendDB |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatExtendTempWrite | ExtendTempWrite |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatFullCompare | FullCompare |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatGetData | GetData |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatIdleCheck | IdleCheck |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatIdleChkpt | IdleChkpt |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatIdleChkTime | IdleChkTime |

| OID | 型 | 名前 | 統計情報 |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatIdleWrite | IdleWrite |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatIndAdd | IndAdd |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatIndLookup | IndLookup |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatIOToRecover | IOToRecover |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatInternal | Internal |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatInternal | Internal |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatLockTablePages | LockTablePages |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatInternal | Internal |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatMaxIO | MaxIO |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatMaxRead | MaxRead |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatMaxWrite | MaxWrite |
| 1.3.6.1.4.1.897.2.2.1.*n* | Counter64 | dbStatPageRelocations | PageRelocations |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatProcedurePages | ProcedurePages |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatQueryCachePages | QueryCachePages |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatQueryLowMemoryStrategy | QueryLowMemoryStrategy |
| 1.3.6.1.4.1.897.2.2.1.*n* | Counter64 | dbStatQueryRowsMaterialized | QueryRowsMaterialized |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatRecoveryUrgency | RecoveryUrgency |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatLogFreeCommit | LogFreeCommit |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatLogWrite | LogWrite |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatRelocatableHeapPages | RelocatableHeapPages |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatRollbackLogPages | RollbackLogPages |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| OID | 型 | 名前 | 統計情報 |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatTempTablePages | TempTablePages |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatTriggerPages | TriggerPages |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatViewPages | ViewPages |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatVersionStorePages | VersionStorePages |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatSnapshotCount | SnapshotCount |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatLockCount | LockCount |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatCacheReadWorkTable | CacheReadWorkTable |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatDiskReadWorkTable | DiskReadWorkTable |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatPrepares | Prepares |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatConnPoolCachedCount | ConnPoolCachedCount |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatConnPoolHits | ConnPoolHits |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatConnPoolMisses | ConnPoolMisses |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatHttpConnPoolCachedCount | HttpConnPoolCachedCount |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatHttpConnPoolHits | HttpConnPoolHits |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatHttpConnPoolMisses | HttpConnPoolMisses |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatHttpConnPoolSteals | HttpConnPoolSteals |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatMirrorServerWaits | MirrorServerWaits |
| 1.3.6.1.4.1.897.2.2.1.*n* | Counter64 | dbStatBytesReceived | BytesReceived |
| 1.3.6.1.4.1.897.2.2.1.*n* | Counter64 | dbStatBytesReceivedUncomp | BytesReceivedUncomp |

| OID | 型 | 名前 | 統計情報 |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.1.*n* | Counter64 | dbStatBytesSent | BytesSent |
| 1.3.6.1.4.1.897.2.2.1.*n* | Counter64 | dbStatBytesSentUncomp | BytesSentUncomp |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatCarverHeapPages | CarverHeapPages |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatClientStmtCacheHits | ClientStmtCacheHits |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatClientStmtCacheMisses | ClientStmtCacheMisses |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatCommit | Commit |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatCursor | Cursor |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatHeapsCarver | HeapsCarver |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatHeapsLocked | HeapsLocked |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatHeapsQuery | HeapsQuery |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatHeapsRelocatable | HeapsRelocatable |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatCursorOpen | CursorOpen |
| 1.3.6.1.4.1.897.2.2.1.*n* | Counter64 | dbStatPacketsReceived | PacketsReceived |
| 1.3.6.1.4.1.897.2.2.1.*n* | Counter64 | dbStatPacketsReceivedUncomp | PacketsReceivedUncomp |
| 1.3.6.1.4.1.897.2.2.1.*n* | Counter64 | dbStatPacketsSent | PacketsSent |
| 1.3.6.1.4.1.897.2.2.1.*n* | Counter64 | dbStatPacketsSentUncomp | PacketsSentUncomp |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatQueryHeapPages | QueryHeapPages |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatQueryMemActiveCurr | QueryMemActiveCurr |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatQueryMemGrantFailed | QueryMemGrantFailed |

| OID | 型 | 名前 | 統計情報 |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatQueryMemGrantGranted | QueryMemGrantGranted |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatQueryMemGrantRequested | QueryMemGrantRequested |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatQueryMemGrantWaited | QueryMemGrantWaited |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatQueryMemGrantWaiting | QueryMemGrantWaiting |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatRequestsReceived | RequestsReceived |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatRlbk | Rlbk |
| 1.3.6.1.4.1.897.2.2.1.*n* | Integer32 | dbStatPrepStmt | PrepStmt |

**参照**


- 「データベースサーバープロパティ値のアクセス」753 ページ
- 「データベースプロパティ値のアクセス」781 ページ


## SQL Anywhere MIB データベースプロパティ

次の表は、SQL Anywhere SNMP Extension Agent を使用して検索できるデータベースプロパティの OID と名前を示します。

書き込み可能なプロパティには、アスタリスク記号 (*) が付いています。値 *n* は、*sasnmp.ini* ファイル内のデータベース番号です。

**説明**

| OID | 型 | 名前 | プロパティ |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.2.1.*n* | String | dbPropAlias | Alias |
| 1.3.6.1.4.1.897.2.2.2.2.*n* | String | dbPropAuditingTypes | AuditingTypes |
| 1.3.6.1.4.1.897.2.2.2.3.*n* | String | dbPropBlankPadding | BlankPadding |

| OID | 型 | 名前 | プロパティ |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.2.4.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.5.*n* | String | dbPropCapabilities | Capabilities |
| 1.3.6.1.4.1.897.2.2.2.6.*n* | String | dbPropCaseSensitive | CaseSensitive |
| 1.3.6.1.4.1.897.2.2.2.7.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.8.*n* | String | dbPropCharSet | CharSet |
| 1.3.6.1.4.1.897.2.2.2.9.*n* | String | dbPropChecksum | Checksum |
| 1.3.6.1.4.1.897.2.2.2.10.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.11.*n* | String | dbPropCollation | Collation |
| 1.3.6.1.4.1.897.2.2.2.12.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.13.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.14.*n* | String | dbPropCurrentRedoPos | CurrentRedoPos |
| 1.3.6.1.4.1.897.2.2.2.15.*n* | String | dbPropDBFileFragments | DBFileFragments |
| 1.3.6.1.4.1.897.2.2.2.16.*n* | String | dbPropDriveType | DriveType |
| 1.3.6.1.4.1.897.2.2.2.17.*n* | String | dbPropEncryption | Encryption |
| 1.3.6.1.4.1.897.2.2.2.18.*n* | String | dbPropFile | File |
| 1.3.6.1.4.1.897.2.2.2.19.*n* | String | dbPropFileSize | FileSize |

| OID | 型 | 名前 | プロパティ |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.2.20.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.21.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.22.*n* | String | dbPropFreePages | FreePages |
| 1.3.6.1.4.1.897.2.2.2.23.*n* | String | dbPropGlobalDBID | GlobalDBID |
| 1.3.6.1.4.1.897.2.2.2.24.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.25.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.26.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.27.*n* | String | dbPropIQStore | IQStore |
| 1.3.6.1.4.1.897.2.2.2.28.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.29.*n* | String | dbPropLanguage | Language |
| 1.3.6.1.4.1.897.2.2.2.30.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.31.*n* | String | dbPropLogFileFragments | LogFileFragments |
| 1.3.6.1.4.1.897.2.2.2.32.*n* | String | dbPropLogMirrorName | LogMirrorName |
| 1.3.6.1.4.1.897.2.2.2.33.*n* | String | dbPropLogName | LogName |
| 1.3.6.1.4.1.897.2.2.2.34.*n* | String | dbPropLTMGeneration | LTMGeneration |
| 1.3.6.1.4.1.897.2.2.2.35.*n* | String | dbPropLTMTrunc | LTMTrunc |

| OID | 型 | 名前 | プロパティ |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.2.36.*n* | String | dbPropMultiByteCharSet | MultiByteCharSet |
| 1.3.6.1.4.1.897.2.2.2.37.*n* | String | dbPropName | Name |
| 1.3.6.1.4.1.897.2.2.2.38.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.39.*n* | String | dbPropPageSize | PageSize |
| 1.3.6.1.4.1.897.2.2.2.40.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.41.*n* | String | dbPropProcedureProfiling* | ProcedureProfiling |
| 1.3.6.1.4.1.897.2.2.2.42.*n* | String | dbPropReadOnly | ReadOnly |
| 1.3.6.1.4.1.897.2.2.2.43.*n* | String | dbPropRemoteTrunc | RemoteTrunc |
| 1.3.6.1.4.1.897.2.2.2.44.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.45.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.46.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.47.*n* | String | dbPropSyncTrunc | SyncTrunc |
| 1.3.6.1.4.1.897.2.2.2.48.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.49.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.50.*n* | String | dbPropTempFileName | TempFileName |
| 1.3.6.1.4.1.897.2.2.2.51.*n* | String | dbPropInternal | Internal |

| OID | 型 | 名前 | プロパティ |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.2.52.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.53.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.54.*n* | String | dbPropNextScheduleTime | NextScheduleTime |
| 1.3.6.1.4.1.897.2.2.2.55.*n* | String | dbPropIdentitySignature | IdentitySignature |
| 1.3.6.1.4.1.897.2.2.2.56.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.57.*n* | String | dbPropSnapshotIsolationState | SnapshotIsolationState |
| 1.3.6.1.4.1.897.2.2.2.58.*n* | String | dbPropConnsDisabled | ConnsDisabled |
| 1.3.6.1.4.1.897.2.2.2.59.*n* | String | dbPropPartnerState | PartnerState |
| 1.3.6.1.4.1.897.2.2.2.60.*n* | String | dbPropArbiterState | ArbiterState |
| 1.3.6.1.4.1.897.2.2.2.61.*n* | String | dbPropMirrorState | MirrorState |
| 1.3.6.1.4.1.897.2.2.2.62.*n* | String | dbPropAlternateServerName | AlternateServerName |
| 1.3.6.1.4.1.897.2.2.2.63.*n* | String | dbPropEncryptionScope | EncryptionScope |
| 1.3.6.1.4.1.897.2.2.2.64.*n* | String | dbPropNcharCharSet | NcharCharSet |
| 1.3.6.1.4.1.897.2.2.2.65.*n* | String | dbPropNcharCollation | NcharCollation |
| 1.3.6.1.4.1.897.2.2.2.66.*n* | String | dbPropAccentSensitive | AccentSensitive |
| 1.3.6.1.4.1.897.2.2.2.67.*n* | String | dbPropSendingTracingTo | SendingTracingTo |

| OID | 型 | 名前 | プロパティ |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.2.68.*n* | String | dbPropReceivingTracingFrom | ReceivingTracingFrom |
| 1.3.6.1.4.1.897.2.2.2.69.*n* | String | dbPropIOParallelism | IOParallelism |
| 1.3.6.1.4.1.897.2.2.2.70.*n* | String | dbPropJavaVM | JavaVM |
| 1.3.6.1.4.1.897.2.2.2.71.*n* | String | dbPropDatabaseCleaner | DatabaseCleaner |
| 1.3.6.1.4.1.897.2.2.2.72.*n* | String | dbPropHasCollationTailoring | HasCollationTailoring |
| 1.3.6.1.4.1.897.2.2.2.73.*n* | String | dbPropCatalogCollation | CatalogCollation |
| 1.3.6.1.4.1.897.2.2.2.74.*n* | String | dbPropHasEndianSwapFix | HasEndianSwapFix |
| 1.3.6.1.4.1.897.2.2.2.75.*n* | String | dbPropAlternateMirrorServerName | AlternateMirrorServerName |
| 1.3.6.1.4.1.897.2.2.2.76.*n* | String | dbPropOptionWatchList | OptionWatchList |
| 1.3.6.1.4.1.897.2.2.2.77.*n* | String | dbPropOptionWatchAction | OptionWatchAction |
| 1.3.6.1.4.1.897.2.2.2.78.*n* | String | dbPropMirrorMode | MirrorMode |
| 1.3.6.1.4.1.897.2.2.2.79.*n* | String | dbPropHasNCHARLegacyCollationFix | HasNCHARLegacyCollationFix |
| 1.3.6.1.4.1.897.2.2.2.80.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.81.*n* | String | dbPropAuthenticated | Authenticated |
| 1.3.6.1.4.1.897.2.2.2.82.*n* | String | dbPropSynchronizationSchemaChangeActive | SynchronizationSchemaChangeActive |
| 1.3.6.1.4.1.897.2.2.2.83.*n* | String | dbPropLastCheckpointTime | LastCheckpointTime |

| OID | 型 | 名前 | プロパティ |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.2.84.*n* | String | dbPropMirrorServerState | MirrorServerState |
| 1.3.6.1.4.1.897.2.2.2.85.*n* | String | dbPropDriveBus | DriveBus |
| 1.3.6.1.4.1.897.2.2.2.86.*n* | String | dbPropDriveModel | DriveModel |
| 1.3.6.1.4.1.897.2.2.2.87.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.88.*n* | String | dbPropMirrorRole | MirrorRole |
| 1.3.6.1.4.1.897.2.2.2.89.*n* | String | dbPropWriteChecksum | WriteChecksum |
| 1.3.6.1.4.1.897.2.2.2.90.*n* | String | dbPropApproximateCPUTime | ApproximateCPUTime |
| 1.3.6.1.4.1.897.2.2.2.91.*n* | String | dbPropConnectedTime | ConnectedTime |
| 1.3.6.1.4.1.897.2.2.2.92.*n* | String | dbPropReqCountUnscheduled | ReqCountUnscheduled |
| 1.3.6.1.4.1.897.2.2.2.93.*n* | String | dbPropReqCountActive | ReqCountActive |
| 1.3.6.1.4.1.897.2.2.2.94.*n* | String | dbPropReqCountBlockIO | ReqCountBlockIO |
| 1.3.6.1.4.1.897.2.2.2.95.*n* | String | dbPropReqCountBlockLock | ReqCountBlockLock |
| 1.3.6.1.4.1.897.2.2.2.96.*n* | String | dbPropReqCountBlockContention | ReqCountBlockContention |
| 1.3.6.1.4.1.897.2.2.2.97.*n* | String | dbPropReqTimeUnscheduled | ReqTimeUnscheduled |
| 1.3.6.1.4.1.897.2.2.2.98.*n* | String | dbPropReqTimeActive | ReqTimeActive |
| 1.3.6.1.4.1.897.2.2.2.99.*n* | String | dbPropReqTimeBlockIO | ReqTimeBlockIO |

| OID | 型 | 名前 | プロパティ |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.2.100.*n* | String | dbPropReqTimeBlockLock | ReqTimeBlockLock |
| 1.3.6.1.4.1.897.2.2.2.101.*n* | String | dbPropReqTimeBlockContention | ReqTimeBlockContention |
| 1.3.6.1.4.1.897.2.2.2.102.*n* | String | dbPropInternal | Internal |
| 1.3.6.1.4.1.897.2.2.2.103.*n* | String | dbPropLastCommitRedoPos | LastCommitRedoPos |
| 1.3.6.1.4.1.897.2.2.2.104.*n* | String | dbPropLastWrittenRedoPos | LastWrittenRedoPos |
| 1.3.6.1.4.1.897.2.2.2.105.*n* | String | dbPropLastSyncedRedoPos | LastSyncedRedoPos |

**参照**

- 「データベースプロパティ値のアクセス」781 ページ

## SQL Anywhere MIB データベースオプション

次の表は、SQL Anywhere SNMP Extension Agent を使用して検索できるデータベースオプションの OID と名前を示します。

書き込み可能なオプションには、アスタリスク記号 (*) が付いています。値 *n* は、*sasnmp.ini* ファイル内のデータベース番号です。

データベースオプションの詳細については、「アルファベット順のオプションリスト」589 ページを参照してください。

**説明**

| OID | 型 | 名前 | オプション |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.3.1.*n* | String | dbOptAllowNullsByDefault* | allow_nulls_by_default |
| 1.3.6.1.4.1.897.2.2.3.2.*n* | String | dbOptAnsinull* | ansinull |
| 1.3.6.1.4.1.897.2.2.3.3.*n* | String | dbOptAnsiBlanks* | ansi_blanks |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| OID | 型 | 名前 | オプション |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.3.4.*n* | String | dbOptAnsiCloseCursorsOnRollback* | ansi_close_cursors_on_rollback |
| 1.3.6.1.4.1.897.2.2.3.5.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.6.*n* | String | dbOptAnsiPermissions* | ansi_permissions |
| 1.3.6.1.4.1.897.2.2.3.7.*n* | String | dbOptAnsiUpdateConstraints* | ansi_update_constraints |
| 1.3.6.1.4.1.897.2.2.3.8.*n* | String | dbOptAuditing* | auditing |
| 1.3.6.1.4.1.897.2.2.3.9.*n* | String | dbOptAuditingOptions* | auditing_options |
| 1.3.6.1.4.1.897.2.2.3.10.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.11.*n* | String | dbOptBackgroundPriority* | background_priority |
| 1.3.6.1.4.1.897.2.2.3.12.*n* | String | dbOptBlocking* | blocking |
| 1.3.6.1.4.1.897.2.2.3.13.*n* | Integer32 | dbOptBlockingTimeout* | blocking_timeout |
| 1.3.6.1.4.1.897.2.2.3.14.*n* | String | dbOptChained* | chained |
| 1.3.6.1.4.1.897.2.2.3.15.*n* | Integer32 | dbOptCheckpointTime* | checkpoint_time |
| 1.3.6.1.4.1.897.2.2.3.16.*n* | Integer32 | dbOptCisOption* | cis_option |
| 1.3.6.1.4.1.897.2.2.3.17.*n* | Integer32 | dbOptCisRowsetSize* | cis_rowset_size |
| 1.3.6.1.4.1.897.2.2.3.18.*n* | String | dbOptCloseOnEndtrans* | close_on_endtrans |
| 1.3.6.1.4.1.897.2.2.3.19.*n* | String | dbOptConnectionAuthentication* | connection_authentication |

| OID | 型 | 名前 | オプション |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.3.20.*n* | String | dbOptContinueAfterRaiserror* | continue_after_raiserror |
| 1.3.6.1.4.1.897.2.2.3.21.*n* | String | dbOptConversionError* | conversion_error |
| 1.3.6.1.4.1.897.2.2.3.22.*n* | String | dbOptCooperativeCommits* | cooperative_commits |
| 1.3.6.1.4.1.897.2.2.3.23.*n* | Integer32 | dbOptCooperativeCommitTimeout* | cooperative_commit_timeout |
| 1.3.6.1.4.1.897.2.2.3.24.*n* | String | dbOptDatabaseAuthentication* | database_authentication |
| 1.3.6.1.4.1.897.2.2.3.25.*n* | String | dbOptDateFormat* | date_format |
| 1.3.6.1.4.1.897.2.2.3.26.*n* | String | dbOptDateOrder* | date_order |
| 1.3.6.1.4.1.897.2.2.3.27.*n* | String | dbOptDebugMessages* | debug_messages |
| 1.3.6.1.4.1.897.2.2.3.28.*n* | String | dbOptDedicatedTask* | dedicated_task |
| 1.3.6.1.4.1.897.2.2.3.29.*n* | Integer32 | dbOptDefaultTimestampIncrement* | default_timestamp_increment |
| 1.3.6.1.4.1.897.2.2.3.30.*n* | String | dbOptDelayedCommits* | delayed_commits |
| 1.3.6.1.4.1.897.2.2.3.31.*n* | Integer32 | dbOptDelayedCommitTimeout* | delayed_commit_timeout |
| 1.3.6.1.4.1.897.2.2.3.32.*n* | String | dbOptDivideByZeroError* | divide_by_zero_error |
| 1.3.6.1.4.1.897.2.2.3.33.*n* | String | dbOptEscapeCharacter* | escape_character |
| 1.3.6.1.4.1.897.2.2.3.34.*n* | String | dbOptExcludeOperators* | exclude_operators |
| 1.3.6.1.4.1.897.2.2.3.35.*n* | String | dbOptExtendedJoinSyntax* | extended_join_syntax |

| OID | 型 | 名前 | オプション |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.3.36.*n* | String | dbOptFireTriggers* | fire_triggers |
| 1.3.6.1.4.1.897.2.2.3.37.*n* | Integer32 | dbOptFirstDayOfWeek* | first_day_of_week |
| 1.3.6.1.4.1.897.2.2.3.38.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.39.*n* | String | dbOptForceViewCreation* | force_view_creation |
| 1.3.6.1.4.1.897.2.2.3.40.*n* | String | dbOptForXmlNullTreatment* | for_xml_null_treatment |
| 1.3.6.1.4.1.897.2.2.3.41.*n* | Integer32 | dbOptGlobalDatabaseId* | global_database_id |
| 1.3.6.1.4.1.897.2.2.3.42.*n* | Integer32 | dbOptIsolationLevel* | isolation_level |
| 1.3.6.1.4.1.897.2.2.3.43.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.44.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.45.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.46.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.47.*n* | String | dbOptLockRejectedRows* | lock_rejected_rows |
| 1.3.6.1.4.1.897.2.2.3.48.*n* | String | dbOptLoginMode* | login_mode |
| 1.3.6.1.4.1.897.2.2.3.49.*n* | String | dbOptLoginProcedure* | login_procedure |
| 1.3.6.1.4.1.897.2.2.3.50.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.51.*n* | Integer32 | dbOptInternal | Internal |

| OID | 型 | 名前 | オプション |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.3.52.*n* | Integer32 | dbOptMaxCursorCount* | max_cursor_count |
| 1.3.6.1.4.1.897.2.2.3.53.*n* | Integer32 | dbOptMaxHashSize* | max_hash_size |
| 1.3.6.1.4.1.897.2.2.3.54.*n* | Integer32 | dbOptMaxPlansCached* | max_plans_cached |
| 1.3.6.1.4.1.897.2.2.3.55.*n* | Integer32 | dbOptMaxRecursiveIterations* | max_recursive_iterations |
| 1.3.6.1.4.1.897.2.2.3.56.*n* | Integer32 | dbOptMaxStatementCount* | max_statement_count |
| 1.3.6.1.4.1.897.2.2.3.57.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.58.*n* | Integer32 | dbOptMinPasswordLength* | min_password_length |
| 1.3.6.1.4.1.897.2.2.3.59.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.60.*n* | Integer32 | dbOptNearestCentury* | nearest_century |
| 1.3.6.1.4.1.897.2.2.3.61.*n* | String | dbOptNonKeywords* | non_keywords |
| 1.3.6.1.4.1.897.2.2.3.62.*n* | String | dbOptOdbcDistinguishCharAndVarchar* | odbc_distinguish_char_and_varchar |
| 1.3.6.1.4.1.897.2.2.3.63.*n* | String | dbOptOnCharsetConversionFailure* | on_charset_conversion_failure |
| 1.3.6.1.4.1.897.2.2.3.64.*n* | String | dbOptOnTsqlError* | on_tsql_error |
| 1.3.6.1.4.1.897.2.2.3.65.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.66.*n* | String | dbOptOptimizationGoal* | optimization_goal |
| 1.3.6.1.4.1.897.2.2.3.67.*n* | Integer32 | dbOptOptimizationLevel* | optimization_level |

| OID | 型 | 名前 | オプション |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.3.68.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.69.*n* | String | dbOptOptimizationWorkload* | optimization_workload |
| 1.3.6.1.4.1.897.2.2.3.70.*n* | Integer32 | dbOptPinnedCursorPercentOfCache* | pinned_cursor_percent_of_cache |
| 1.3.6.1.4.1.897.2.2.3.71.*n* | Integer32 | dbOptPrecision* | precision |
| 1.3.6.1.4.1.897.2.2.3.72.*n* | String | dbOptPrefetch* | prefetch |
| 1.3.6.1.4.1.897.2.2.3.73.*n* | String | dbOptPreserveSourceFormat* | preserve_source_format |
| 1.3.6.1.4.1.897.2.2.3.74.*n* | String | dbOptPreventArticlePkeyUpdate* | prevent_article_pkey_update |
| 1.3.6.1.4.1.897.2.2.3.75.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.76.*n* | String | dbOptQuotedIdentifier* | quoted_identifier |
| 1.3.6.1.4.1.897.2.2.3.77.*n* | String | dbOptReadPastDeleted* | read_past_deleted |
| 1.3.6.1.4.1.897.2.2.3.78.*n* | Integer32 | dbOptRecoveryTime* | recovery_time |
| 1.3.6.1.4.1.897.2.2.3.79.*n* | String | dbOptReplicateAll* | replicate_all |
| 1.3.6.1.4.1.897.2.2.3.80.*n* | String | dbOptReturnDateTimeAsString* | return_date_time_as_string |
| 1.3.6.1.4.1.897.2.2.3.81.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.82.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.83.*n* | String | dbOptRowCounts* | row_counts |

| OID | 型 | 名前 | オプション |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.3.84.*n* | Integer32 | dbOptScale* | scale |
| 1.3.6.1.4.1.897.2.2.3.85.*n* | String | dbOptSortCollation* | sort_collation |
| 1.3.6.1.4.1.897.2.2.3.86.*n* | String | dbOptSqlFlaggerErrorLevel* | sql_flagger_error_level |
| 1.3.6.1.4.1.897.2.2.3.87.*n* | String | dbOptSqlFlaggerWarningLevel* | sql_flagger_warning_level |
| 1.3.6.1.4.1.897.2.2.3.88.*n* | String | dbOptStringRtruncation* | string_rtruncation |
| 1.3.6.1.4.1.897.2.2.3.89.*n* | String | dbOptSubsumeRowLocks* | subsume_row_locks |
| 1.3.6.1.4.1.897.2.2.3.90.*n* | String | dbOptSuppressTdsDebugging* | suppress_tds_debugging |
| 1.3.6.1.4.1.897.2.2.3.91.*n* | String | dbOptTdsEmptyStringIsNull* | tds_empty_string_is_null |
| 1.3.6.1.4.1.897.2.2.3.92.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.93.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.94.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.95.*n* | String | dbOptTimestampFormat* | timestamp_format |
| 1.3.6.1.4.1.897.2.2.3.96.*n* | String | dbOptTimeFormat* | time_format |
| 1.3.6.1.4.1.897.2.2.3.97.*n* | Integer32 | dbOptTimeZoneAdjustment* | time_zone_adjustment |
| 1.3.6.1.4.1.897.2.2.3.98.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.99.*n* | String | dbOptTruncateTimestampValues* | truncate_timestamp_values |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

| OID | 型 | 名前 | オプション |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.3.100.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.101.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.102.*n* | String | dbOptTsqlVariables* | tsql_variables |
| 1.3.6.1.4.1.897.2.2.3.103.*n* | String | dbOptUpdateStatistics* | update_statistics |
| 1.3.6.1.4.1.897.2.2.3.104.*n* | String | dbOptUpgradeDatabaseCapability* | upgrade_database_capability |
| 1.3.6.1.4.1.897.2.2.3.105.*n* | String | dbOptUserEstimates* | user_estimates |
| 1.3.6.1.4.1.897.2.2.3.106.*n* | String | dbOptWaitForCommit* | wait_for_commit |
| 1.3.6.1.4.1.897.2.2.3.107.*n* | String | dbOptTempSpaceLimitCheck* | temp_space_limit_check |
| 1.3.6.1.4.1.897.2.2.3.108.*n* | Integer32 | dbOptRemoteIdleTimeout* | remote_idle_timeout |
| 1.3.6.1.4.1.897.2.2.3.109.*n* | String | dbOptAnsiSubstring* | ansi_substring |
| 1.3.6.1.4.1.897.2.2.3.110.*n* | String | dbOptOdbcDescribeBinaryAsVarbinary* | odbc_describe_binary_as_varbinary |
| 1.3.6.1.4.1.897.2.2.3.111.*n* | String | dbOptRollbackOnDeadlock* | rollback_on_deadlock |
| 1.3.6.1.4.1.897.2.2.3.112.*n* | String | dbOptIntegratedServerName* | integrated_server_name |
| 1.3.6.1.4.1.897.2.2.3.113.*n* | String | dbOptLogDeadlocks* | log_deadlocks |
| 1.3.6.1.4.1.897.2.2.3.114.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.115.*n* | String | dbOptWebserviceNamespaceHost* | webservice_namespace_host |

| OID | 型 | 名前 | オプション |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.3.116.*n* | Integer32 | dbOptMaxQueryTasks* | max_query_tasks |
| 1.3.6.1.4.1.897.2.2.3.117.*n* | Integer32 | dbOptRequestTimeout* | request_timeout |
| 1.3.6.1.4.1.897.2.2.3.118.*n* | String | dbOptSynchronizeMirrorOnCommit* | synchronize_mirror_on_commit |
| 1.3.6.1.4.1.897.2.2.3.119.*n* | Integer32 | dbOptHttpSessionTimeout* | http_session_timeout |
| 1.3.6.1.4.1.897.2.2.3.120.*n* | String | dbOptUuidHasHyphens* | uuid_has_hyphens |
| 1.3.6.1.4.1.897.2.2.3.121.*n* | String | dbOptAllowSnapshotIsolation* | allow_snapshot_isolation |
| 1.3.6.1.4.1.897.2.2.3.122.*n* | String | dbOptVerifyPasswordFunction* | verify_password_function |
| 1.3.6.1.4.1.897.2.2.3.123.*n* | String | dbOptDefaultDbspace* | default_dbspace |
| 1.3.6.1.4.1.897.2.2.3.124.*n* | String | dbOptCollectStatisticsOnDmlUpdates* | collect_statistics_on_dml_updates |
| 1.3.6.1.4.1.897.2.2.3.125.*n* | String | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.126.*n* | String | dbOptJavaLocation* | java_location |
| 1.3.6.1.4.1.897.2.2.3.127.*n* | String | dbOptOemString* | oem_string |
| 1.3.6.1.4.1.897.2.2.3.128.*n* | Integer32 | dbOptMaxTempSpace* | max_temp_space |
| 1.3.6.1.4.1.897.2.2.3.129.*n* | String | dbOptSecureFeatureKey* | secure_feature_key |
| 1.3.6.1.4.1.897.2.2.3.130.*n* | String | dbOptMaterializedViewOptimization* | materialized_view_optimization |
| 1.3.6.1.4.1.897.2.2.3.131.*n* | Integer32 | dbOptUpdatableStatementIsolation* | updatable_statement_isolation |

| OID | 型 | 名前 | オプション |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.3.132.*n* | String | dbOptTsqlOuterJoins* | tsql_outer_joins |
| 1.3.6.1.4.1.897.2.2.3.133.*n* | String | dbOptPostLoginProcedure* | post_login_procedure |
| 1.3.6.1.4.1.897.2.2.3.134.*n* | String | dbOptConnAuditing* | conn_auditing |
| 1.3.6.1.4.1.897.2.2.3.135.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.136.*n* | String | dbOptJavaVmOptions* | java_vm_options |
| 1.3.6.1.4.1.897.2.2.3.137.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.138.*n* | Integer32 | dbOptMaxClientStatementsCached* | max_client_statements_cached |
| 1.3.6.1.4.1.897.2.2.3.139.*n* | String | dbOptQueryMemTimeout* | query_mem_timeout |
| 1.3.6.1.4.1.897.2.2.3.140.*n* | String | dbOptAllowReadClientFile* | allow_read_client_file |
| 1.3.6.1.4.1.897.2.2.3.141.*n* | String | dbOptAllowWriteClientFile* | allow_write_client_file |
| 1.3.6.1.4.1.897.2.2.3.142.*n* | String | dbOptPriority* | priority |
| 1.3.6.1.4.1.897.2.2.3.143.*n* | String | dbOptMaxPriority* | max_priority |
| 1.3.6.1.4.1.897.2.2.3.144.*n* | String | dbOptProgressMessages* | progress_messages |
| 1.3.6.1.4.1.897.2.2.3.145.*n* | Integer32 | dbOptBlockingOthersTimeout* | blocking_others_timeout |
| 1.3.6.1.4.1.897.2.2.3.146.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.147.*n* | String | dbOptTimestampWithTimeZoneFormat* | timestamp_with_time_zone_format |

| OID | 型 | 名前 | オプション |
|---|---|---|---|
| 1.3.6.1.4.1.897.2.2.3.148.*n* | Integer32 | dbOptHttpConnectionPoolTimeout* | http_connection_pool_timeout |
| 1.3.6.1.4.1.897.2.2.3.149.*n* | Integer32 | dbOptHttpConnectionPoolBasesize* | http_connection_pool_basesize |
| 1.3.6.1.4.1.897.2.2.3.150.*n* | String | dbOptStGeometryDescribeType* | st_geometry_describe_type |
| 1.3.6.1.4.1.897.2.2.3.151.*n* | String | dbOptStGeometryAstextFormat* | st_geometry_astext_format |
| 1.3.6.1.4.1.897.2.2.3.152.*n* | String | dbOptStGeometryAsbinaryFormat* | st_geometry_asbinary_format |
| 1.3.6.1.4.1.897.2.2.3.153.*n* | String | dbOptStGeometryAsxmlFormat* | st_geometry_asxml_format |
| 1.3.6.1.4.1.897.2.2.3.154.*n* | String | dbOptReservedKeywords* | reserved_keywords |
| 1.3.6.1.4.1.897.2.2.3.155.*n* | String | dbOptStGeometryOnInvalid* | st_geometry_on_invalid |
| 1.3.6.1.4.1.897.2.2.3.156.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.157.*n* | Integer32 | dbOptInternal | Internal |
| 1.3.6.1.4.1.897.2.2.3.158.*n* | String | dbOptJavaClassPath* | java_class_path |
| 1.3.6.1.4.1.897.2.2.3.159.*n* | String | dbOptStGeometryInterpolation* | st_geometry_interpolation |

# RDBMS MIB リファレンス

以下の項では、SQL Anywhere SNMP Extension Agent を使用して検索できる値の OID を示します。デフォルトで、RDBMS MIB は *C:¥Program Files¥SQL Anywhere 12¥snmp¥RDBMS-MIB.mib* にあります。

## rdbmsDbTable

このテーブルは、システムにインストールされているデータベースについての情報を示します。

値 *db* は、*sasnmp.ini* ファイル内のデータベース番号です。

| OID | 型 | 名前 | 戻り値 |
|---|---|---|---|
| 1.3.6.1.2.1.39.1.1.1.1.*db* | Integer | rdbmsDbIndex | *db* |
| 1.3.6.1.2.1.39.1.1.1.2.*db* | OID | rdbmsDbPrivateMibOID | 1.3.6.1.4.1.897.2 |
| 1.3.6.1.2.1.39.1.1.1.3.*db* | String | rdbmsDbVendorName | PROPERTY( 'CompanyName' ) |
| 1.3.6.1.2.1.39.1.1.1.4.*db* | String | rdbmsDbName | DB_PROPERTY( 'Name' ) |
| 1.3.6.1.2.1.39.1.1.1.5.*db* | String | rdbmsDbContact | PROPERTY( 'LicensedUser' ) |

## rdbmsDbInfoTable

このテーブルは、システム上のデータベースについての追加情報を示します。

値 *db* は、*sasnmp.ini* ファイル内のデータベース番号です。

| OID | 型 | 名前 | 戻り値 |
|---|---|---|---|
| 1.3.6.1.2.1.39.1.2.1.1.*db* | String | rdbmsDbInfoProductName | PROPERTY('ProductName') |
| 1.3.6.1.2.1.39.1.2.1.2.*db* | String | rdbmsDbInfoVersion | PROPERTY('ProductVersion') |
| 1.3.6.1.2.1.39.1.2.1.3.*db* | Integer | rdbmsDbInfoSizeUnits | dbInfoSizeAllocated と dbInfoSizeUsed に基づいて計算される<br>●1= バイト<br>●2=KB<br>●3=MB<br>●4=GB<br>●5=TB<br>(各単位はその 1 つ前の単位の 1024 倍) |

| OID | 型 | 名前 | 戻り値 |
|---|---|---|---|
| 1.3.6.1.2.1.39.1.2.1.4.*db* | Integer | rdbmsDbInfoSizeAllocated | DB_PROPERTY('PageSize') * DB_PROPERTY('FileSize') |
| 1.3.6.1.2.1.39.1.2.1.5.*db* | Integer | rdbmsDbInfoSizeUsed | DB_PROPERTY('PageSize') * (DB_PROPERTY('FileSize') - DB_PROPERTY('FreePages') ) |
| 1.3.6.1.2.1.39.1.2.1.6.*db* | String | rdbmsDbInfoLastBackup | NULL[1] |

[1] この OID は SQL Anywhere SNMP Extension Agent ではサポートされていません。

## rdbmsDbParamTable

このテーブルは、システム上のデータベースの設定パラメーターを示します。

値 *db* は *sasnmp.ini* ファイル内のデータベース番号です。また、*n* は sa.2.3 サブツリー内のオプションのインデックスです。

| OID | 型 | 名前 | 戻り値 |
|---|---|---|---|
| 1.3.6.1.2.1.39.1.3.1.1.*db* | String | rdbmsDbParamName | オプション名 |
| 1.3.6.1.2.1.39.1.3.1.2.*db* | Integer | rdbmsDbParamSubIndex | *n* |
| 1.3.6.1.2.1.39.1.3.1.3.*db* | OID | rdbmsDbParamID | このオプションに対応する SQL Anywhere MIB 内の OID |
| 1.3.6.1.2.1.39.1.3.1.4.*db* | String | rdbmsDbParamCurrValue | オプション値 |
| 1.3.6.1.2.1.39.1.3.1.5.*db* | String | rdbmsDbParamComment | NULL[1] |

[1] この OID は SQL Anywhere SNMP Extension Agent ではサポートされていません。

## rdbmsDbLimitedResourceTable

このテーブルは、各 DB 領域の空き領域についての情報を示します。表中の *n* は、以下に示すとおり、いずれかの DB 領域を表す値になります。

- 1 ～ 13：通常の DB 領域 (データベース内では番号 0 ～ 12)
- 14：トランザクションログファイル
- 15：トランザクションログミラーファイル

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

●16：テンポラリファイル

値 *db* は、*sasnmp.ini* ファイル内のデータベース番号です。

| OID | 型 | 名前 | 戻り値 |
|---|---|---|---|
| 1.3.6.1.2.1.39.1.4.1.1.*n.db* | String | rdbmsDbLimitedResourceName | DB 領域の名前、トランザクションログ、トランザクションログミラー、またはテンポラリファイル |
| 1.3.6.1.2.1.39.1.4.1.2.*n.db* | OID | rdbmsDbLimitedResourceID | 1.3.6.1.4.1.897.2 |
| 1.3.6.1.2.1.39.1.4.1.3.*n.db* | Integer | rdbmsDbLimitedResourceLimit | ディスク上の使用可能な空き領域 + 現在のファイルサイズ |
| 1.3.6.1.2.1.39.1.4.1.4.*n.db* | Integer | rdbmsDbLimitedResourceCurrent | 現在のファイルサイズ |
| 1.3.6.1.2.1.39.1.4.1.5.*n.db* | Integer | rdbmsDbLimitedResourceHighwater | 現在のサイズ |
| 1.3.6.1.2.1.39.1.4.1.6.*n.db* | Integer | rdbmsDbLimitedResourceFailure | 0[1] |
| 1.3.6.1.2.1.39.1.4.1.7.*n.db* | String | rdbmsDbLimitedResourceDescription | バイト、KB、MB、GB、TB のいずれか |

[1] この OID は SQL Anywhere SNMP Extension Agent ではサポートされていません。

## rdbmsSrvTable

このテーブルは、システム上の動作中またはインストール済みのデータベースサーバーを示します。

値 *db* は、*sasnmp.ini* ファイル内のデータベース番号です。

| OID | 型 | 名前 | 戻り値 |
|---|---|---|---|
| 1.3.6.1.2.1.39.1.5.1.1.*db* | OID | rdbmsSrvPrivateMibOID | 1.3.6.1.4.1.897.2 |
| 1.3.6.1.2.1.39.1.5.1.2.*db* | String | rdbmsSrvVendorName | PROPERTY( 'CompanyName' ) |
| 1.3.6.1.2.1.39.1.5.1.3.*db* | String | rdbmsSrvProductName | PROPERTY( 'ProductName' ) |

| OID | 型 | 名前 | 戻り値 |
|---|---|---|---|
| 1.3.6.1.2.1.39.1.5.1.4.*db* | String | rdbmsSrvContact | PROPERTY( 'LicensedCompany' ) |

## rdbmsSrvInfoTable

このテーブルは、システム内のデータベースサーバーについての追加情報を示します。

値 *db* は、*sasnmp.ini* ファイル内のデータベース番号です。

| OID | 型 | 名前 | 戻り値 |
|---|---|---|---|
| 1.3.6.1.2.1.39.1.6.1.1.*db* | Integer | rdbmsSrvInfoStartupTime | PROPERTY( 'StartTime' ) |
| 1.3.6.1.2.1.39.1.6.1.2.*db* | Integer | rdbmsSrvInfoFinishedTransactions | 0[1] |
| 1.3.6.1.2.1.39.1.6.1.3.*db* | Integer | rdbmsSrvInfoDiskReads | PROPERTY( 'DiskReadEng' ) |
| 1.3.6.1.2.1.39.1.6.1.4.*db* | Integer | rdbmsSrvInfoLogicalReads | 0[1] |
| 1.3.6.1.2.1.39.1.6.1.5.*db* | Integer | rdbmsSrvInfoDiskWrites | PROPERTY( 'DiskWriteEng' ) |
| 1.3.6.1.2.1.39.1.6.1.6.*db* | Integer | rdbmsSrvInfoLogicalWrites | 0[1] |
| 1.3.6.1.2.1.39.1.6.1.7.*db* | Integer | rdbmsSrvInfoPageReads | 0[1] |
| 1.3.6.1.2.1.39.1.6.1.8.*db* | Integer | rdbmsSrvInfoPageDiskOutOfWrites | 0[1] |
| 1.3.6.1.2.1.39.1.6.1.9.*db* | Integer | rdbmsSrvInfoSpaces | 0[1] |
| 1.3.6.1.2.1.39.1.6.1.10.*db* | Integer | rdbmsSrvInfoHandledRequests | PROPERTY( 'Req' ) |
| 1.3.6.1.2.1.39.1.6.1.11.*db* | Integer | rdbmsSrvInfoRequestRecvs | PROPERTY( 'PacketsReceivedUncomp' ) |
| 1.3.6.1.2.1.39.1.6.1.12.*db* | Integer | rdbmsSrvInfoRequestSends | PROPERTY( 'PacketsSentUncomp' ) |
| 1.3.6.1.2.1.39.1.6.1.13.*db* | Integer | rdbmsSrvInfoHighwaterInboundAssociations | 0[1] |
| 1.3.6.1.2.1.39.1.6.1.14.*db* | Integer | rdbmsSrvInfoMaxInboundAssociations | 0[1] |

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

[1] この OID は SQL Anywhere SNMP Extension Agent ではサポートされていません。

## rdbmsSrvParamTable

このテーブルは、SQL Anywhere SNMP Extension Agent で SQL Anywhere MIB を使用して設定できるサーバーオプションを示します。*n* はインデックスです。このインデックスの詳細は次のとおりです。

| *n* | サーバーオプション |
|---|---|
| 1 | ConnsDisabled |
| 2 | LivenessTimeout (デフォルト) |
| 3 | QuittingTime |
| 4 | RememberLastStatement |
| 5 | RequestLogFile |
| 6 | RequestLogging |

値 *db* は、*sasnmp.ini* ファイル内のデータベース番号です。

| OID | 型 | 名前 | 戻り値 |
|---|---|---|---|
| 1.3.6.1.2.1.39.1.7.1.1.*n.db* | String | rdbmsDbSrvParamName | オプション *n* の名前 |
| 1.3.6.1.2.1.39.1.7.1.2.*n.db* | Integer | rdbmsDbSrvParamSubIndex | *n* |
| 1.3.6.1.2.1.39.1.7.1.3.*n.db* | OID | rdbmsDbSrvParamID | 1.3.6.1.4.1.897.2 |
| 1.3.6.1.2.1.39.1.7.1.4.*n.db* | String | rdbmsDbSrvParamCurrValue | オプション *n* の現在の値 |
| 1.3.6.1.2.1.39.1.7.1.5.*n.db* | String | rdbmsDbSrvParamComment | オプション *n* の完全名 |

## rdbmsSrvLimitedResourceTable

このテーブルには、サーバー設定パラメーターについての情報が格納されています。

値 *db* は、*sasnmp.ini* ファイル内のデータベース番号です。また、*n* はリソースのインデックスです。このインデックスの詳細は次のとおりです。

| *n* | 名前 | リソース | リソースの制限 |
|---|---|---|---|
| 1 | Connections | PROPERTY( 'UniqueClientAddresses' ) | PROPERTY('LicenseCount') |
| 2 | Processors | PROPERTY('NumLogicalProcessors Used') | PROPERTY('NumLogicalProcessorsUsed') |

| OID | 型 | 名前 | 戻り値 |
|---|---|---|---|
| 1.3.6.1.2.1.39.1.8.1.1.*db* | String | rdbmsSrvLimitedResource Name | リソース *n* の名前 |
| 1.3.6.1.2.1.39.1.8.1.2.*db* | OID | rdbmsSrvLimitedResource ID | このオプションに対応する SQL Anywhere MIB 内の OID |
| 1.3.6.1.2.1.39.1.8.1.3.*db* | Integer | rdbmsSrvLimitedResource Limit | リソース *n* の上限 |
| 1.3.6.1.2.1.39.1.8.1.4.*db* | Integer | rdbmsSrvLimitedResource Current | リソース *n* の現在の値 |
| 1.3.6.1.2.1.39.1.8.1.5.*db* | Integer | rdbmsSrvLimitedResource Highwater | リソース *n* の現在の値 |
| 1.3.6.1.2.1.39.1.8.1.6.*db* | Integer | rdbmsSrvLimitedResource Failures | 0[1] |
| 1.3.6.1.2.1.39.1.8.1.7.*db* | String | rdbmsSrvLimitedResource Description | リソース *n* の名前 |

[1] この OID は SQL Anywhere SNMP Extension Agent ではサポートされていません。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

# セキュリティ

この項では、SQL Anywhere のセキュリティ機能について説明します。

## データのセキュリティ

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

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

- **ユーザー ID と認証**　データベースにアクセスするユーザーを制御します。「新しいユーザーの作成」513 ページを参照してください。

- **任意アクセス制御機能**　データベースへの接続中にユーザーが実行するアクションを制御します。「ユーザー ID、権限、パーミッション」512 ページを参照してください。

- **監査**　データベースで行われたアクションの記録を管理するのに役立ちます。「データベースアクティビティの監査」1388 ページを参照してください。

- **データベースサーバーオプション**　データベースのロードなどの管理作業を実行するユーザーを指定します。このオプションは、データベースサーバーの起動時に設定されます。「コマンドラインからパーミッションを制御する」159 ページを参照してください。

- **ビューとストアドプロシージャー**　ユーザーがアクセスするデータとユーザーが実行する操作を指定します。「高度なセキュリティを実現するためのビューとプロシージャー」564 ページを参照してください。

- **データベースとテーブルの暗号化**　データベースを安全に管理するために、単純暗号化または強力な暗号化のいずれかを選択できます。単純暗号化は、難読化と同じです。強力な暗号化にすると、暗号化キーなしではデータベースにまったくアクセスできなくなります。「-ek dbeng12/dbsrv12 データベースオプション」310 ページと「DatabaseKey (DBKEY) 接続パラメーター」335 ページを参照してください。

  テーブル暗号化機能では、データベース全体ではなく個々のテーブルを暗号化できます。「テーブル暗号化」1405 ページを参照してください。

- **トランスポートレイヤーセキュリティ**　トランスポートレイヤーセキュリティを使用すると、クライアントアプリケーションとデータベースサーバー間の通信を認証することができます。トランスポートレイヤーセキュリティでは、楕円曲線暗号方式または RSA 暗号方式を使用します。「トランスポートレイヤーセキュリティ」1410 ページを参照してください。

> **注意**
> データベースサーバーを実行しているコンピューター上で他のプロセスがクライアント／サーバー通信の内容にアクセスできるようになることが心配な場合は、暗号化の使用をおすすめします。

> **注意**
> 別途ライセンスが必要な必須コンポーネント
>
> ECC 暗号化と FIPS 認定の暗号化には、別途ライセンスが必要です。強力な暗号化テクノロジはすべて、輸出規制対象品目です。
>
> 「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

- **機能の保護**　データベースサーバー上のすべてのデータベースで機能を無効にすることができます。

- **SELinux のサポート**　SELinux のポリシーを使用して、アプリケーションのシステムリソースへのアクセスを制御します。SQL Anywhere には、Red Hat Enterprise Linux 5 で SQL Anywhere を保護するポリシーが含まれています。

  SQL Anywhere の SELinux ポリシーのコンパイルとインストールについては、*$SQLANY12/selinux/readme* を参照してください。

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

ユーザー ID とパーミッションはセキュリティ関連のトピックです。「ユーザー ID、権限、パーミッション」512 ページを参照してください。

# セキュリティに関するヒント

データベース管理者は、データのセキュリティを強化するためにさまざまなアクションを実行できます。次に例を示します。

- **パスワードの慎重な選択**　デフォルトのユーザー ID とパスワードを使用するデータベースは、決して配布しないでください。「パスワードのセキュリティ」1383 ページを参照してください。

- **DBA 権限の制限**　DBA 権限は非常に強力であるため、本当に必要なユーザーにだけ付与してください。DBA 権限を持っているユーザーは、データベース内で何でも参照したり実行したりできます。

  データベース管理者に 2 つのユーザー ID を提供することを検討してください。1 つを DBA 権限付き、もう 1 つを DBA 権限なしにすれば、必要なときにだけ DBA 権限を持つユーザーとして接続できます。

- **保護されたデータベース機能の使用**　-sf データベースサーバーオプションでは、データベースサーバー上のすべてのデータベースに対して機能を有効または無効にすることができます。無効にできる機能には、外部ストアドプロシージャーや Java の使用、リモートデータアクセス、要求ログ設定の変更機能などがあります。「-sf dbeng12/dbsrv12 サーバーオプション」271 ページと「機能の保護」1387 ページを参照してください。


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

- **外部システム関数の削除**　次の外部関数は、セキュリティ上のリスクを生じさせる危険性があります。

  ○「xp_cmdshell システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』

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

  ○「xp_startmail システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』

  ○「xp_startsmtp システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』

  ○「xp_sendmail システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』

  ○「xp_stopmail システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』

  ○「xp_stopsmtp システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』

  電子メールコマンドを使用すると、ユーザーはサーバーに自分が作成した電子メールを送信させることができます。悪意のあるユーザーであれば、電子メールコマンドやコマンドシェルプロシージャーを使用して、付与されていない権限でオペレーティングシステムのタスクを実行することもありえます。セキュリティを考慮した環境では、このような関数は削除してください。

  プロシージャーの削除については、「DROP PROCEDURE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

- **データベースファイルの保護**　データベースファイル、ログファイル、DB 領域ファイルを不正アクセスから保護する必要があります。このようなファイルは共有ディレクトリまたはボリュームには保管しないでください。

- **データベースソフトウェアの保護**　SQL Anywhere ソフトウェアを、不正アクセスから保護する必要があります。ユーザーには、アプリケーション、DLL、その他の必要なリソースへのアクセス権だけを付与してください。

- **サービスまたはデーモンとしてのデータベースサーバーの実行**　権限のないユーザーがサーバーを停止したり、データベースやログファイルへのアクセスを取得したりしないように、データベースサーバーを Windows サービスとして実行してください。UNIX では、同様の目的でサーバーをデーモンとして実行します。「現在のセッション外でのデータベースサーバーの起動」170 ページを参照してください。

- **SATMP 環境変数をユニークなディレクトリに設定 (UNIX)**　UNIX プラットフォームでデータベースサーバーを保護するには、SATMP をユニークなディレクトリに設定し、他のすべてのユーザーに対して、このディレクトリの読み取り、書き込み、実行を制限します。これによって、すべての接続が強制的に TCP/IP を使用することになり、共有メモリ接続よりも安全になります。

  クライアントとサーバー間で使用される共有メモリのバッファーは、実際のデータが 2 つのサイト間で送受信される前に、ディレクトリツリーから削除されます。これは、共有メモリのバッファーやファイルが非表示であるため、他のプロセスは通信データを参照できず、データを処理できないことを意味します。「SATMP 環境変数」451 ページを参照してください。

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

  詳細については、「-ep dbeng12/dbsrv12 サーバーオプション」226 ページと「-ek dbeng12/dbsrv12 データベースオプション」310 ページを参照してください。

# データベースアクセス

データベース管理者は、ユーザー ID とパスワードを割り当てることによって、どのユーザーがデータベースにアクセスするかを制御します。各ユーザー ID にパーミッションを付与することによって、データベース接続しているときに各ユーザーが実行できるタスクを制御します。

### ユーザー ID に基づくパーミッションのスキーム

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

- ユーザーが作成したオブジェクト
- ユーザーに対して明示的なパーミッションが付与されているオブジェクト
- ユーザーの所属グループに明示的なパーミッションが付与されているオブジェクト

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

詳細については、次の項を参照してください。

- 「ユーザー ID、権限、パーミッション」512 ページ
- 「CONNECT 文 [ESQL] [Interactive SQL]」『SQL Anywhere サーバー SQL リファレンス』
- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「REVOKE 文」『SQL Anywhere サーバー SQL リファレンス』

### 統合化ログインの使用方法

統合化ログインを使用すると、ユーザーは 1 つのログイン名とパスワードで Windows オペレーティングシステムとデータベースの両方にログインできます。外部ログイン名は、データベースユーザー ID に関連付けられています。統合化ログインを行う場合、ユーザーはログイン名とパスワードの両方を指定してオペレーティングシステムにログオンします。オペレーティングシステムはそのユーザーをサーバーに通知し、サーバーは関連付けられたデータベースユーザー ID としてそのユーザーをログインさせます。追加のログイン名とパスワードは必要ありません。

統合化ログインを使用する場合、ユーザープロファイル Guest でブランクのパスワードを使用できるようにしておくと、統合化ログインを受け入れるサーバーで管理しているデータベースへのアクセスが無制限に許可されます。ユーザーはデフォルトではログイン時に Guest ユーザーのプロファイルを使用するため、現実には、どのユーザーでも任意のログイン ID やパスワードを使用してサーバーにログインできてしまいます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

詳細については、次の項を参照してください。


- 「セキュリティについての考慮事項：無制限データベースアクセス」60 ページ
- 「Windows 統合ログイン」51 ページ
- 「login_mode オプション」635 ページ


## パスワードのセキュリティ

パスワードは、データベースのセキュリティシステムの重要な部分です。安全のために、パスワードは容易に推測できないものにし、ハードドライブやその他のロケーションから簡単にアクセスできないようにしてください。SQL Anywhere のパスワードでは、常に大文字と小文字が区別されます。パスワード認証に使用する関数を verify_password_function オプションで指定できます。「verify_password_function オプション」708 ページを参照してください。

### ログインポリシーの実装

ログインポリシーを使用して、パスワードの変更頻度の設定や、アカウントがロックされるまでに実行可能なログイン試行回数の指定を行います。「ログインポリシー」553 ページまたは「CREATE LOGIN POLICY 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

### デフォルトのユーザー ID とパスワードの変更

新しく作成されるデータベースのデフォルトのユーザー ID とパスワードは、**DBA** と **sql** です。このパスワードを変更してから、データベースを展開してください。

### 最小長のパスワードの実装

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

```
SET OPTION PUBLIC.min_password_length = 8;
```

「min_password_length オプション」648 ページを参照してください。

### パスワード有効期限の導入

デフォルトでは、データベースパスワードに有効期限はありません。ログインポリシーを使用して、パスワードの有効期限を導入できます。「ログインポリシー」553 ページを参照してください。

### ODBC データソースにパスワードを含めない

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

ODBC データソースを作成するとき、または Sybase Central 接続プロファイルを作成するときに、オプションでパスワードを含めることができます。権限のないユーザーによって参照されないようにするため、パスワードは含めないようにしてください。

「ODBC データソース」36 ページを参照してください。

### パスワードを含む設定ファイルの暗号化

設定ファイルを作成するとき、パスワード情報をオプションとして組み込むことができます。パスワードを保護するために、ファイル難読化ユーティリティ (dbfhide) を使用し、単純暗号化で設定ファイルの内容を隠すことを検討してください。「ファイル非表示ユーティリティ (dbfhide)」953 ページを参照してください。

### パスワード検証の使用

verify_password_function オプションを使用して、パスワード規則を実装する関数を指定できます。「verify_password_function オプション」708 ページを参照してください。

次の例では、テーブルと関数を 1 つずつ定義し、いくつかのログインポリシーオプションを設定しています。これらは共に、パスワードに特定の種類の文字が含まれることを要求し、パスワードの再利用を禁止して、パスワード有効期限を適用するなど、詳細なパスワード規則を実装します。ユーザー ID が作成されるか、パスワードが変更されると、データベースサーバーによって verify_password_function オプションを使用して関数が呼び出されます。アプリケーションは、post_login_procedure オプションで指定されたプロシージャーを呼び出して、パスワードが期限切れになる前にパスワードの変更が必要であることを通知できます。

このサンプルコードは、*%SQLANYSAMP12%¥SQLAnywhere¥SQL¥verify_password.sql* でも利用できます。

```
-- This example defines a function that implements advanced password rules
-- including requiring certain types of characters in the password and
-- disallowing password reuse. The f_verify_pwd function is called by the
-- server using the verify_password_function option when a user ID is
-- created or a password is changed.
--
-- The "root" login profile is configured to expire passwords every 180 days
-- and lock non-DBA accounts after 5 consecutive failed login attempts.
--
-- The application may call the procedure specified by the
-- post_login_procedure option to report that the password should be changed
-- before it expires.


-- only DBA should have permissions on this table
CREATE TABLE DBA.t_pwd_history(
      pk         INT        DEFAULT AUTOINCREMENT PRIMARY KEY,
      user_name   CHAR(128),         -- the user whose password is set
      pwd_hash    CHAR(32) );        -- hash of password value to detect
                                     -- duplicate passwords

-- called whenever a non-NULL password is set
-- to verify the password conforms to password rules
CREATE FUNCTION DBA.f_verify_pwd( uid     VARCHAR(128),
                                  new_pwd VARCHAR(255) )
RETURNS VARCHAR(255)
BEGIN
   -- a table with one row per character in new_pwd
   DECLARE local temporary table pwd_chars(
         pos INT PRIMARY KEY,    -- index of c in new_pwd
         c   CHAR( 1 CHAR ) );   -- character
   -- new_pwd with non-alpha characters removed
   DECLARE pwd_alpha_only      CHAR(255);
```

```
   DECLARE num_lower_chars     INT;

   -- enforce minimum length (can also be done with
   -- min_password_length option)
   IF length( new_pwd ) < 6 THEN
      RETURN 'password must be at least 6 characters long';
   END IF;

   -- break new_pwd into one row per character
   INSERT INTO pwd_chars SELECT row_num, substr( new_pwd, row_num, 1 )
                         FROM dbo.RowGenerator
                         WHERE row_num <= length( new_pwd );

   -- copy of new_pwd containing alpha-only characters
   SELECT list( c, '' ORDER BY pos ) INTO pwd_alpha_only
      FROM pwd_chars WHERE c BETWEEN 'a' AND 'z' OR c BETWEEN 'A' AND 'Z';

   -- number of lowercase characters IN new_pwd
   SELECT count(*) INTO num_lower_chars
      FROM pwd_chars WHERE CAST( c AS BINARY ) BETWEEN 'a' AND 'z';

   -- enforce rules based on characters contained in new_pwd
   IF ( SELECT count(*) FROM pwd_chars WHERE c BETWEEN '0' AND '9' )
         < 1 THEN
      RETURN 'password must contain at least one numeric digit';
   ELSEIF length( pwd_alpha_only ) < 2 THEN
      RETURN 'password must contain at least two letters';
   ELSEIF num_lower_chars = 0
         OR length( pwd_alpha_only ) - num_lower_chars = 0 THEN
      RETURN 'password must contain both upper- and lowercase characters';
   END IF;

   -- not the same as any user name
   -- (this could be modified to check against a disallowed words table)
   IF EXISTS( SELECT * FROM SYS.SYSUSER
                WHERE lower( user_name ) IN ( lower( pwd_alpha_only ),
                                              lower( new_pwd ) ) ) THEN
      RETURN 'password or only alphabetic characters in password ' ||
            'must not match any user name';
   END IF;

   -- not the same as any previous password for this user
   IF EXISTS( SELECT * FROM t_pwd_history
                WHERE user_name = uid
                  AND pwd_hash = hash( uid || new_pwd, 'md5' ) ) THEN
      RETURN 'previous passwords cannot be reused';
   END IF;

   -- save the new password
   INSERT INTO t_pwd_history( user_name, pwd_hash )
      VALUES( uid, hash( uid || new_pwd, 'md5' ) );

   RETURN( NULL );
END;

ALTER FUNCTION DBA.f_verify_pwd SET HIDDEN;
GRANT EXECUTE ON DBA.f_verify_pwd TO PUBLIC;
SET OPTION PUBLIC.verify_password_function = 'DBA.f_verify_pwd';


-- All passwords expire in 180 days.  Expired passwords can be changed
-- by the user using the NewPassword connection parameter.
ALTER LOGIN POLICY root password_life_time = 180;
```

```
-- If an application calls the procedure specified by the
-- post_login_procedure option, then the procedure can be used to warn
-- the user that their password is about to expire.  In particular,
-- Interactive SQL and Sybase Central call the
-- post_login_procedure system procedure.
ALTER LOGIN POLICY root password_grace_time = 30;

-- Five consecutive failed login attempts will result in a non-DBA
-- user ID being locked.
ALTER LOGIN POLICY root max_failed_login_attempts = 5;
```

## ユーザーが実行できるタスクの制御

パーミッションを付与することによって、ユーザーがデータベースオブジェクトに対して実行できるタスク (作成、変更、実行、更新など) を制御できます。権限の付与によって、ユーザーが実行できる管理タスク (バックアップ、プロファイリングなど) を制御できます。

パーミッションと権限を付与するには、GRANT 文を使用します。パーミッションについては、オブジェクトに対する権限を付与するパーミッションを、他のユーザーに委任することもできます。

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

**参照**

- 「ユーザー ID、権限、パーミッション」512 ページ
- 「GRANT 文」『SQL Anywhere サーバー SQL リファレンス』
- 「REVOKE 文」『SQL Anywhere サーバー SQL リファレンス』

## データベースオブジェクトのセキュリティ

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

**参照**

- 「プロシージャー、トリガー、ユーザー定義関数の利点」『SQL Anywhere サーバー SQL の使用法』
- 「高度なセキュリティを実現するためのビューとプロシージャー」564 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

# 機能の保護

ユーザーが使用できるデータベース機能を制限するために、データベースサーバーの起動時に機能保護オプション (-sf) を指定できます。機能保護オプションでは、次のような機能の使用可能性を指定できます。

- サーバー側のバックアップ
- 外部ストアドプロシージャー
- リモートデータアクセス
- Web サービス

機能の完全なリストについては、「-sf dbeng12/dbsrv12 サーバーオプション」271 ページを参照してください。

データベースサーバーを起動するときには、-sk オプションを指定することもできます。このオプションでは、特定の接続で保護されている機能を有効にするキーを指定します。特定の接続で保護されている機能を有効にするには、secure_feature_key テンポラリオプションをデータベースサーバーの起動時に -sk で指定された値に設定します。

sa_server_option システムプロシージャーを使用して、無効になっている機能または機能セットに変更を加えるためには、-sk でキーを指定し、secure_feature_key テンポラリオプションをそのキーの値に設定する必要があります。機能を無効または有効にするために行った変更は、直ちに有効になります。

**♦ データベース機能の保護**

1. -sf オプションを指定してデータベースサーバーを起動します。必要に応じて -sk オプションも指定します。

   たとえば、次のコマンドは、データベースサーバーを起動し、リモートデータアクセスの使用を無効にします。ただし、無効になっている機能を有効にするためのキーも指定されています。

   ```
   dbsrv12 -n secure_server -sf remote_data_access -sk ls64uwq15 c:¥mydata.db
   ```

2. データベースに接続します。

   次に例を示します。

   ```
   dbisql -c "UID=DBA;PWD=sql;Host=myhost;Server=secure_server;DBN=demo"
   ```

3. secure_feature_key テンポラリオプションをデータベースサーバーの起動時に -sk で指定された値に設定します。

   次に例を示します。

   ```
   SET TEMPORARY OPTION secure_feature_key = 'ls64uwq15';
   ```

4. sa_server_option システムプロシージャーを使用して、このデータベースサーバーで保護されている機能を変更します。

次に例を示します。

```
CALL sa_server_option( 'SecureFeatures', '-remote_data_access' );
```

**参照**

- 「-sf dbeng12/dbsrv12 サーバーオプション」271 ページ
- 「-sk dbeng12/dbsrv12 サーバーオプション」275 ページ
- 「secure_feature_key」677 ページ
- 「sa_server_option システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』

# データベースアクティビティの監査

各データベースには、関連付けられたトランザクションログファイルがあります。トランザクションログはデータベースのリカバリに使用します。これは、データベースに対して実行されたトランザクションの記録です。「トランザクションログ」127 ページを参照してください。

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

「監査」は、データベース上で行われたアクティビティをトラッキングする方法です。監査を使用すると、追加のデータがトランザクションログに保存されます。

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

データベースで監査が有効になっている場合、トランザクションログの使用を停止することはできません。トランザクションログをオフにするためには、先に監査をオフにする必要があります。

## 監査の制御

データベース管理者が監査を有効にすると、セキュリティ関連の情報がトランザクションログに追加されます。これには、Sybase Central または Interactive SQL を使用します。

**♦ 監査の制御 (Sybase Central の場合)**

監査はデフォルトでは無効になっています。監査を有効または無効にするには、DBA 権限が必要です。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

1. **SQL Anywhere 12** プラグインを使用して、DBA 権限のあるユーザーとして、データベースに接続します。

2. データベースを右クリックし、**[プロパティ]** をクリックします。

3. **[監査]** タブをクリックし、次のいずれかを選択します。
   - **[このデータベースの監査情報を収集しない]**　監査情報は収集されません。このオプションを選択すると、auditing データベースオプションが Off になり、監査が無効になります。「auditing オプション」598 ページを参照してください。
   - **[このデータベースの監査情報をすべて収集する]**　データベースのすべてのタイプの監査情報を収集します。このオプションを選択すると、auditing データベースオプションが On になり、監査が有効になります。「auditing オプション」598 ページを参照してください。

     このオプションを選択すると、トランザクションログが大幅に増大する可能性があります。
   - **[このデータベースの次のタイプの監査情報を収集する]**　収集する監査情報を指定できます。たとえば、DDL の変更のみを収集するように選択できます。「sa_enable_auditing_type システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

     このオプションを選択すると、auditing_options データベースオプションの設定が変更されます。「auditing_options オプション」599 ページを参照してください。

4. **[OK]** をクリックします。

**♦ 監査の制御 (Interactive SQL の場合)**

1. DBA 権限のあるユーザーとしてデータベースに接続します。

2. 監査を有効にするには次の文を実行します。

   ```
   SET OPTION PUBLIC.auditing = 'On';
   ```

   有効にする監査情報のタイプを指定するには、次のシステムプロシージャーを使用します。

   ```
   CALL sa_enable_auditing_type( 'all' );
   ```

   **all** を、有効にする監査のタイプに置き換えることによって、収集する監査情報のタイプを制御できます。「sa_enable_auditing_type システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

3. 監査を無効にするには次の文を実行します。

   ```
   SET OPTION PUBLIC.auditing = 'Off';
   ```

   無効にする監査情報のタイプを指定するには、次のシステムプロシージャーを使用します。

   ```
   CALL sa_disable_auditing_type( 'all' );
   ```

**all** を、無効にする監査のタイプに置き換えることによって、特定タイプの監査情報を収集の対象から除外できます。「sa_disable_auditing_type システムプロシージャー」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

### 個々の接続の監査

データベースの監査を有効にした後で、データベースログインプロシージャーに conn_auditing database テンポラリオプションを指定し、接続ごとに監査を有効にすることができます。クライアントコンピューターの IP アドレスや接続のタイプなどの情報に基づいて監査を有効にできます。

ログインプロシージャーに conn_auditing オプションを設定しなかった場合、このオプションはデフォルトで On に設定されます。

次の例は、DBA ユーザーによる接続を除くすべての接続で監査を有効にするログインプロシージャーの一部を示しています。

```
DECLARE usr VARCHAR(128)
SELECT CONNECTION_PROPERTY( 'Userid' ) INTO usr;
IF usr != 'DBA' THEN
  SET TEMPORARY OPTION conn_auditing='On'
ELSE
  SET TEMPORARY OPTION conn_auditing='Off'
END IF;
```

### 参照

- 「login_procedure オプション」636 ページ
- 「conn_auditing オプション」607 ページ
- 「auditing オプション」598 ページ

## 監査情報の取り出し

Sybase Central またはログ変換ユーティリティ (dbtran) を使用して、トランザクションログから監査情報を取り出すことができます。

**♦ 監査情報を取り出すには、次の手順に従います (Sybase Central の場合)。**

1. バックアップ権限または DBA、Remote、または Backup 権限を持つユーザーでデータベースに接続します。
2. データベースを選択します。
3. **[監査]** タブをクリックします。
4. **[監査メッセージの取得]** をクリックします。

   ウィンドウが開き、dbtran メッセージが表示されます。年代順の出力に関する警告は無視します。
5. **[閉じる]** をクリックします。

右ウィンドウ枠の **[監査]** タブに監査情報が表示されます。

6. フィルターオプションを使用して、表示する監査情報を制御します。

   すべての監査情報を表示するか、エラーのみを表示するか、指定したテキストを含む監査メッセージのみを表示するかを選択できます。

7. 監査エントリテーブルでエントリを選択して、そのエントリの詳細を表示します。

8. 最新の監査情報を取り出すには、[F5] キーを押してからこの手順を繰り返します。

詳細については、「監査の例」1392 ページを参照してください。

### dbtran ユーティリティを使用した監査情報の取り出し

dbtran ユーティリティには、Sybase Central またはコマンドプロンプトでアクセスできます。dbtran ユーティリティは指定のトランザクションログを使用して、すべてのトランザクションと、各コマンドを実行したユーザーの情報を保持する SQL スクリプトを生成します。-g オプションを使用すると、dbtran は監査情報を含むコメントをより多く含むようになります。-g オプションを指定すると、次のオプションを指定した場合と同じ結果になります。

- **-d** 出力を年代順に表示する
- **-t** トリガーで生成したオペレーションを出力に含める
- **-a** ロールバックトランザクションを出力に含める

これらのオプションの詳細については、「ログ変換ユーティリティ (dbtran)」985 ページを参照してください。

dbtran ユーティリティは、稼働中のデータベースサーバーに対して、またはデータベースログファイルに対して実行できます。

#### ♦ 稼働中のデータベースサーバーからの監査情報の取り出し

監査情報を取り出す前に、DBA、REMOTE、BACKUP のいずれかの権限を持つユーザーとしてデータベースに接続していることを確認してください。

- データベースサーバーが稼動している状態で、次のコマンドを実行します。

  ```
  dbtran -g -c connection-string -n SQL-file
  ```

  次に例を示します。

  ```
  dbtran -g -c "UID=DBA;PWD=sql" -n demo.sql
  ```

  読み取り可能なバージョンのトランザクションログが、現在のディレクトリに保存されます。この例では、監査情報が *demo.sql* ファイルに保存され、このファイルにはサンプルデータベースに関する情報が含まれます。

  接続文字列の詳細については、「接続パラメーター」320 ページを参照してください。

♦ **トランザクションログファイルからの監査情報の取り出し**

1. データベースサーバーを停止して、トランザクションログファイルが使用可能であることを確認します。

2. 次のコマンドを実行します。

```
dbtran -g transaction-log SQL-file
```

次に例を示します。

```
dbtran -g demo.log demo.sql
```

この例では、トランザクションログファイル *demo.log* から取り出した監査情報が、*demo.sql* に保存されます。

詳細については、「ログ変換ユーティリティ (dbtran)」985 ページを参照してください。

## 監査コメント

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

次に例を示します。

```
CALL sa_audit_string( 'Started audit testing here.' );
```

このコメントは監査文としてトランザクションログに格納されます。

## 監査の例

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

### Sybase Central の監査の例

1. Sybase Central を起動し、SQL Anywhere 12 Demo データソースを使用してサンプルデータベースに接続します。

   DBA 権限のあるユーザーとして接続します。

2. 監査を有効にします。

   a. データベースを右クリックし、**[プロパティ]** をクリックします。

   b. **[監査]** タブをクリックします。

   c. **[このデータベースの監査情報をすべて収集する]** をクリックします。

d. **[適用]** をクリックします。

e. **[OK]** をクリックします。

3. パスワードを welcome にして、Test1 というユーザーをサンプルデータベースに追加します。

   a. **[ユーザーとグループ]** を右クリックし、**[新規]** » **[ユーザー]** を選択します。

   b. プロンプトが表示されたら、ユーザー名に **Test1**、パスワードに **welcome** を入力します。

   c. ユーザーに **[プロファイル]** 権限を付与します。

   d. **[完了]** をクリックします。

   e. サンプルデータベースを切断します。

4. Sybase Central を使用して、Test1 としてサンプルデータベースに接続し、Employees テーブル内の機密情報へのアクセスを試みます。

   a. **[テーブル]** を選択し、Employees テーブルをクリックします。

   b. **[データ]** タブをクリックします。

      "Employees" テーブルから選択するためのパーミッションがないことを示すエラーメッセージが表示されます。「パーミッションがありません : %1」『エラーメッセージ』を参照してください。

   c. **[OK]** をクリックします。

   d. サンプルデータベースを切断します。

5. このアクティビティの監査情報を表示します。

   a. Sybase Central を使用して、DBA 権限を持つユーザーとしてサンプルデータベースに接続します。

   b. データベースを選択し、右ウィンドウ枠の **[監査]** タブをクリックします。

   c. **[監査メッセージの取得]** をクリックします。

   d. **[閉じる]** をクリックします。

      監査情報が表示されます。

   e. フィルタリングオプションを使用して、監査情報テーブル内のエラーを探します。**[エラーのみ]** オプションを選択することによって、Test1 のエラーを見つけることができます。日付と時刻の情報を使用して、エラーを特定します。たとえば、2007 年 11 月 6 日 10:07:14 に Test1 が Employees テーブルへのアクセスを試みた場合、対応する監査エントリは次のようになります。

      2007-11-06 10:07:14 | Permission

6. サンプルデータベースを元の状態にリストアします。

   a. データベースを右クリックし、**[プロパティ]** をクリックします。

   b. **[監査]** タブで、**[このデータベースの監査情報を収集しない]** をクリックします。

   c. **[OK]** をクリックします。

   d. **[ユーザーとグループ]** をクリックします。

   Test1 を右クリックし、**[削除]** をクリックします。

### Interactive SQL の監査の例

1. Interactive SQL を起動し、SQL Anywhere 12 Demo データソースを使用してサンプルデータベースに接続します。

   このとき DBA ユーザーとして接続します。

2. 次のように SET OPTION 文を指定して、監査を有効にします。

```
SET OPTION PUBLIC.auditing = 'On';
```

3. 次のように CREATE USER 文を指定して、サンプルデータベースにユーザー Test1 を追加します。

```
CREATE USER Test1
IDENTIFIED BY welcome;
```

4. 新しい Interactive SQL ウィンドウを開き、Test1 としてサンプルデータベースに接続し、次の SELECT 文を使用して Employees テーブルの機密情報へのアクセスを試みます。

```
SELECT Surname, Salary
  FROM GROUPO.Employees;
```

   "Employees" テーブルから選択するためのパーミッションがないことを示すエラーメッセージが表示されます。「パーミッションがありません : %1」『エラーメッセージ』を参照してください。

5. 次のコマンドを実行して、このアクティビティの監査情報を表示します。

```
dbtran -g -c "DSN=SQL Anywhere 12 Demo" -n demo.sql
```

6. サンプルデータベースを元の状態にリストアします。

   ●DROP USER 文を使用して、データベースから Test1 ユーザーを削除します。

```
DROP USER Test1;
```

   ●次の SET OPTION 文を使用して、監査を無効にします。

```
SET OPTION PUBLIC.auditing = 'Off';
```

## データベースサーバー外のアクションの監査

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

Windows または UNIX でアクションの監査を実行する場合、dbtran または dblog を使用すると、データベースファイルと同じディレクトリに拡張子 *.alg* のテキストファイルが生成されます。たとえば、*demo.db* というデータベースファイルに対しては、*demo.alg* というファイルが生成されます。ツール名、Windows または UNIX ユーザー名、日付／時刻を含むレコードがこのファイルに追加されます。auditing オプションが On に設定されている場合は、レコードが *.alg* ファイルに追加されるだけです。

**参照**


- 「auditing オプション」598 ページ
- 「ログ変換ユーティリティ (dbtran)」985 ページ
- 「トランザクションログユーティリティ (dblog)」1041 ページ


## データベースサーバーの保護

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

- **データベースの開始と停止**　パーソナルデータベースサーバーを使用している場合、デフォルトでは、すべてのユーザーが実行中のサーバー上で追加のデータベースを起動できます。ネットワークデータベースサーバーの場合、デフォルトでは、実行中のデータベースサーバー上で別のデータベースを起動するためには DBA 権限が必要です。-gd オプションは、この機能を実行できるユーザーを、すでに接続しているデータベースで特定のレベルのパーミッションを付与されているユーザーに制限できます。このオプションに指定できる値は、DBA、all、none です。「-gd dbeng12/dbsrv12 サーバーオプション」233 ページを参照してください。

- **データベースの作成と削除**　パーソナルデータベースサーバーを実行している場合、デフォルトでは、すべてのユーザーが CREATE DATABASE 文を使用してデータベースファイルを作成できます。ネットワークデータベースサーバーの場合、デフォルトではデータベースの作成に DBA 権限が必要です。-gu オプションは、この機能を実行できるユーザーを、すでに接続しているデータベースで特定のレベルのパーミッションを付与されているユーザーに制限できます。このオプションに指定できる値は、DBA、all、none、utility_db です。「-gu dbeng12/dbsrv12 サーバーオプション」247 ページを参照してください。

- **サーバーの停止**　dbstop ユーティリティは、データベースサーバーを停止します。このユーティリティは、バッチファイルや、サーバーを対話形式で (データベースサーバーメッセージウィンドウの **[シャットダウン]** をクリックして) 停止できない場合に便利です。パーソナルデータベースサーバーの場合、デフォルトでは、すべてのユーザーが dbstop を実行してサーバーを停止できます。ネットワークデータベースサーバーの場合、デフォルトではデータベースサーバーの停止に DBA 権限が必要です。-gk オプションは、この機能を実行できる

ユーザーを、データベースで特定のレベルのパーミッションを付与されているユーザーに制限できます。このオプションに指定できる値は、DBA、all、none です。「-gk dbeng12/dbsrv12 サーバーオプション」235 ページを参照してください。

- **データのロードとアンロード**　LOAD TABLE 文、UNLOAD TABLE 文、UNLOAD 文はすべて、データベースサーバーコンピューター上のファイルシステムにアクセスできます。UNIX 以外のオペレーティングシステムを使用するパーソナルデータベースサーバーの場合、デフォルト設定は all です。ネットワークデータベースサーバーや UNIX パーソナルサーバーの場合、デフォルト設定は DBA です。パーソナルデータベースサーバーが稼働中の場合、すでにファイルシステムにアクセスできるため、セキュリティ上の問題はありません。ネットワークデータベースサーバーを稼働中の場合は、ファイルシステムへの不当なアクセスによってセキュリティ問題の起こる可能性があります。-gl オプションを使用して、データのロードとアンロードを行うのに必要なデータベースパーミッションを制御できます。このオプションに指定できる値は、DBA、all、none です。「-gl dbeng12/dbsrv12 サーバーオプション」236 ページを参照してください。

- **トランスポートレイヤーセキュリティによるクライアント／サーバー通信の暗号化**　トランスポートレイヤーセキュリティを使用して、クライアントアプリケーションとデータベースサーバー間の通信を認証することで、ネットワークパケットのセキュリティを高めることができます。トランスポートレイヤーセキュリティでは、楕円曲線暗号方式または RSA 暗号方式を使用します。「トランスポートレイヤーセキュリティ」1410 ページを参照してください。

- **データベース機能の無効化**　-sf サーバーオプションを使用すると、データベースサーバー上で実行されているデータベースで無効にする機能のリストを指定できます。このリストに含まれる機能は、クライアントアプリケーションに加え、データベース内に定義されているストアドプロシージャー、トリガー、イベントでも使用できません。この設定が役に立つのは、使用するデータベースが自分の所有しているものではないため、ウイルスやトロイの木馬などの望ましくないアクションが組み込まれている可能性がある場合です。「-sf dbeng12/dbsrv12 サーバーオプション」271 ページを参照してください。

**参照**

- 「セキュリティに関するヒント」1380 ページ

# データベースの暗号化と復号化

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

> **注意**
> データベースが暗号化されている場合、WinZip などのツールでデータベースを圧縮しても、元のデータベースファイルよりも大幅に小さくはなりません。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## 単純暗号化の使用

♦ **単純暗号化の使用**

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

- dbinit -ea simple オプションを使用して、データベースを作成します。

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

  ```
  dbinit -ea simple test.db
  ```

**参照**

- 「初期化ユーティリティ (dbinit)」958 ページ
- 「CREATE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』

## 強力な暗号化

強力なデータベース暗号化方式では、キー (パスワード) がないとデータベースの操作やアクセスを行うことができません。アルゴリズムは、データベースやトランザクションログファイルに含まれる情報をエンコードして解読できないようにしています。

> **警告**
> 強力な暗号化が適用されたデータベースの場合、キーのコピーは必ず安全な場所に保管してください。暗号化キーがわからなくなった場合は、Sybase 製品の保守契約を結んでいるテクニカルサポートに依頼してもデータにはアクセスできません。アクセスできなくなったデータベースは、廃棄して、新しくデータベースを作成する必要があります。

### サポートされている強力な暗号化アルゴリズム

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

AES_FIPS (128 ビット) または AES256_FIPS (256 ビット) タイプを使用することで、別の FIPS 認定の AES アルゴリズムを指定して強力な暗号化を実装することもできます。-fips オプションを指定してデータベースサーバーを起動した場合、AES、AES256、AES_FIPS、または AES256_FIPS の強力な暗号方式で暗号化されたデータベースを実行できますが、単純暗号化方式で暗号化されたデータベースは実行できません。暗号化されていないデータベースはサーバーで開始できます。「-fips dbeng12/dbsrv12 サーバーオプション」230 ページを参照してください。

AES_FIPS または AES256_FIPS で暗号化したデータベースを実行するために使用するコンピューターには、SQL Anywhere セキュリティオプションをインストールしてください。

**注意**
別途ライセンスが必要な必須コンポーネント

ECC 暗号化と FIPS 認定の暗号化には、別途ライセンスが必要です。強力な暗号化テクノロジはすべて、輸出規制対象品目です。

「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

**注意**
FIPS 認定の暗号化は、すべてのプラットフォームで使用できるわけではありません。サポートされるプラットフォームのリストについては、http://www.ianywhere.jp/tech/1061806-os_components.html を参照してください。

### データベースの強力な暗号化の設定の制御

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

既存のデータベースでは強力な暗号化のオンとオフを簡単に切り替えることはできませんが、強力な暗号化を実装するには次の 3 つの方法があります。つまり、強力な暗号化を指定して新規にデータベースを作成するか、既存のデータベースを再構築するときに暗号化ステータスを変更するか、既存のデータベースに対して CREATE ENCRYPTED DATABASE 文を実行することができます。

データベースを再構築して、既存のデータベースに含まれるすべてのデータとスキーマをアンロードできます。新しいデータベースを作成して (ここで強力な暗号化のステータスを含めたさまざまな設定を変更できます)、データを新しいデータベースに再ロードします。強力に暗号化されたデータベースをアンロードするにはキーが必要です。

**参照**

- 「データベースの再ロード」『SQL Anywhere サーバー SQL の使用法』
- 「初期化ユーティリティ (dbinit)」958 ページ
- 「CREATE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「CREATE ENCRYPTED DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』

## 暗号化されているデータベースの作成

暗号化されたデータベースは、次の方法で作成できます。

- dbinit ユーティリティと、強力な暗号化を有効にするための各種オプションの組み合わせ。

  dbinit ユーティリティの -ep オプションと -ek オプションを使用すると、強力な暗号化が適用されたデータベースが作成され、プロンプトボックスまたはコマンドラインで暗号化キーを指定できます。dbinit -ea オプションは、暗号化アルゴリズムを AES または AES256 (FIPS 認定アルゴリズムの場合は AES_FIPS または AES256_FIPS) に設定します。「初期化ユーティリティ (dbinit)」958 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

●Sybase Central の**データベース作成ウィザード**で強力に暗号化されたデータベースを作成する。「データベースの作成 (Sybase Central)」118 ページを参照してください。

●データベースの dbunload ユーティリティを使用し、新規データベースを強力な暗号化で作成するためのオプションを指定する。-an オプションは、新規データベースを作成します。強力な暗号化と暗号化キーをプロンプトボックスまたはコマンドラインで指定するには、-ep オプションまたは -ek オプションを使用します。-ea オプションは、暗号化アルゴリズムを AES または AES256 (FIPS 認定アルゴリズムの場合は AES_FIPS または AES256_FIPS) に設定します。「アンロードユーティリティ (dbunload)」1044 ページを参照してください。

●Sybase Central の**データベースアンロードウィザード**で強力に暗号化されたデータベースを作成する。「データベースアンロードウィザードを使用したデータのエクスポート」『SQL Anywhere サーバー SQL の使用法』を参照してください。

●次の SQL 文を使用する。

○「CREATE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
○「CREATE ENCRYPTED DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
○「CREATE ENCRYPTED FILE 文」『SQL Anywhere サーバー SQL リファレンス』

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

1. Interactive SQL から既存のデータベースに接続します。

2. ENCRYPTION 句、KEY オプション、ALGORITHM オプションを含む CREATE DATABASE 文を実行します。

   たとえば、次の文は、FIPS 認定の 128 ビット AES 暗号化を使用して、*c:¥*ディレクトリにデータベースファイル *myencrypteddb.db* を作成します。

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

**♦ 暗号化されたデータベースを作成するには、次の手順に従います (コマンドラインの場合)。**

1. dbinit ユーティリティを使用してデータベースを作成します。コマンドプロンプトまたはウィンドウで暗号化キーを指定するには、-ek または -ep をそれぞれ指定します。

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

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

2. 次のコマンドを入力して、データベースを起動します。

```
dbeng12 myencrypteddb.db -ek "0kZ2o56AK#"
```

**♦ 既存のデータベースを使用して、暗号化されたデータベースを作成するには、次の手順に従います (SQL の場合)。**

1. Interactive SQL から既存のデータベース (コピーするデータベース以外) に接続します。

2. CREATE ENCRYPTED DATABASE 文を使用してデータベースを暗号化します。

   次の文では、データベースファイル *demo.db* の AES で暗号化されたコピーを作成し、そのコピーに *encryptedDemo.db* という名前を付けます。

   ```
   CREATE ENCRYPTED DATABASE 'encryptedDemo.db'
   FROM 'demo.db'
   KEY 'abc'
   ALGORITHM 'AES';
   ```

   CREATE ENCRYPTED DATABASE 文を実行すると、ファイルが暗号化 (上書き) されるのではなく、ファイルのコピーが暗号化形式で作成されます。データベースに関連付けられたトランザクションログ、トランザクションログミラー、または DB 領域がある場合は、これらのファイルの暗号化されたコピーも作成されます。「CREATE ENCRYPTED DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

### テクニカルサポートのためのデータベースの暗号化

リカバリが必要なデータベースがあり、Sybase 製品の保守契約を結んでいるテクニカルサポートに送信するために暗号化したい場合は、CREATE ENCRYPTED FILE 文を使用します。トランザクションログ、トランザクションログミラー、DB 領域ファイルなどのデータベース関連ファイルもすべてこの文を使用して暗号化します。「CREATE ENCRYPTED FILE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

### CREATE ENCRYPTED DATABASE 文と CREATE ENCRYPTED FILE 文の比較

既存のデータベースを暗号化する場合は CREATE ENCRYPTED DATABASE 文を使用します。CREATE ENCRYPTED FILE 文は、リカバリが必要なデータベースを暗号化する場合にのみ使用します。

どちらの文にも DBA 権限が必要であり、また文を実行するときに暗号化するデータベースに接続していることはできません。

CREATE ENCRYPTED FILE 文と CREATE ENCRYPTED DATABASE 文には次の違いがあります。

- CREATE ENCRYPTED FILE 文はデータベース関連ファイル (トランザクションログ、トランザクションログミラー、DB 領域) ごとに実行する必要がありますが、CREATE ENCRYPTED DATABASE 文では、データベース関連ファイルがすべて自動的に暗号化されます。

- CREATE ENCRYPTED DATABASE 文はリカバリが必要なデータベースには使用できませんが、CREATE ENCRYPTED FILE 文は使用できます。

- CREATE ENCRYPTED DATABASE 文は、プロシージャー、トリガー、またはバッチ内では使用できません。CREATE ENCRYPTED FILE 文は使用できます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

●CREATE ENCRYPTED DATABASE 文では単純暗号化アルゴリズムがサポートされていますが、CREATE ENCRYPTED FILE 文ではこのアルゴリズムはサポートされていません。

**参照**

暗号化キーの詳細については、「DatabaseKey (DBKEY) 接続パラメーター」335 ページを参照してください。

Windows Mobile では、ARM プロセッサー用に AES_FIPS および AES256_FIPS アルゴリズムのみがサポートされています。

> **注意**
> FIPS は、すべてのプラットフォームで使用できるわけではありません。サポートされるプラットフォームのリストについては、http://www.ianywhere.jp/tech/1061806-os_components.html を参照してください。

## データベースの復号化

CREATE DECRYPTED DATABASE 文を使用して、データベースを復号化することができます。この文では、CREATE ENCRYPTED DATABASE 文の場合と同じように、元のデータベースファイルが上書きされるのではなく、ファイルのコピー (この場合は復号化形式) が作成されます。

**♦ データベースを復号化するには、次の手順に従います (SQL の場合)。**

1. Interactive SQL から既存のデータベースに接続します。
2. CREATE DECRYPTED DATABASE 文を使用してデータベースを復号化します。

   最初の文では、*demo.db* の AES256 で暗号化されたコピー *demoEncrypted.db* を作成します。2 番目の文では、*demoEncrypted.db* の復号化されたコピー *demoDecrypted.db* を作成します。

   ```
   CREATE ENCRYPTED DATABASE 'demoEncrypted.db'
     FROM 'demo.db'
     KEY 'Sd8f6654*Mnn'
     ALGORITHM 'AES256';
   CREATE DECRYPTED DATABASE 'demoDecrypted.db'
     FROM 'demoEncrypted.db'
     KEY 'Sd8f6654*Mnn';
   ```

   データベースに関連付けられたトランザクションログ、トランザクションログミラー、または DB 領域がある場合は、これらのファイルの復号化されたコピーも作成されます。「CREATE DECRYPTED DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

### テクニカルサポートのためのデータベースの復号化

リカバリが必要なデータベースがあり、テクニカルサポートに送信するために復号化したい場合は、CREATE DECRYPTED FILE 文を使用します。トランザクションログ、トランザクションログミラー、DB 領域ファイルなどのデータベース関連ファイルもすべてこの文を使用して復号化します。「CREATE DECRYPTED FILE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

# 暗号化キー

暗号化キーには簡単に推測できない値を選択することをおすすめします。キーの長さは任意ですが、短いと推測されやすいため、一般的には長い方が適しています。また、数字、文字、特殊文字を組み合わせると、キーは推測されにくくなります。

暗号化キーでは常に大文字と小文字が区別されます。また、前後のスペースや、セミコロンを含めることはできません。

データベースを起動するたびに、キーを指定してください。キーを忘れた場合はデータベースにまったくアクセスできなくなります。

暗号化キーの入力に、コマンドプロンプト (デフォルト) またはプロンプトボックスのいずれかを選択できます。プロンプトボックスでのキー入力を選択すると、キーが表示されないため、さらにセキュリティが強化されます。クライアントでは、データベースを起動するたびにキーを指定してください。データベース管理者がデータベースを起動する場合は、クライアントでキーを使用する必要はありません。「-ep dbeng12/dbsrv12 サーバーオプション」226 ページを参照してください。

**警告**
強力な暗号化が適用されたデータベースの場合、キーのコピーは必ず安全な場所に保管してください。暗号化キーがわからなくなった場合は、Sybase 製品の保守契約を結んでいるテクニカルサポートに依頼してもデータにはアクセスできません。アクセスできなくなったデータベースは、廃棄して、新しくデータベースを作成する必要があります。

CREATE ENCRYPTED DATABASE 文を使用して、暗号化されたデータベースや、テーブル暗号化が有効になっているデータベースの暗号化キーを変更することができます。データベースを暗号化する場合と同じように、既存のファイルに上書きするのではなく、ファイルのコピーを作成し、そのコピーを新しいキーで暗号化します。

**♦ データベースの暗号化キーを変更するには、次の手順に従います。**

- CREATE ENCRYPTED DATABASE 文を使用して、暗号化されたデータベースの暗号化キーを変更します。

  次の例では、abc というキーで暗号化されたデータベースファイル *myOldDatabase.db* のコピーを作成し、そのコピーに *myNewDatabase.db* という名前を付けて、キー abc123 で暗号化します。他のすべてのデータベース関連ファイル (トランザクションログ、トランザクションログミラー、DB 領域ファイル) も新しい暗号化キーを使用して作成されます。「CREATE ENCRYPTED DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

```
CREATE ENCRYPTED DATABASE myNewDatabase.db
FROM myOldDatabase.db
KEY 'abc123'
OLD KEY 'abc'
ALGORITHM 'AES';
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## パフォーマンスの問題

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

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

**参照**

- 「パフォーマンス向上のためのキャッシュの使用」『SQL Anywhere サーバー SQL の使用法』
- 「-c dbeng12/dbsrv12 サーバーオプション」204 ページ

## データベース部分の暗号化

データベースの一部だけを暗号化する場合は、カラムまたはテーブルを暗号化することを選択できます。

カラムの暗号化は、任意のテーブル内の任意のカラムに対していつでも実行できます。テーブルの暗号化を行うには、データベースでテーブルの暗号化が有効になっている必要があります。テーブルの暗号化は、データベースの作成 (初期化) 時に有効にします。

### カラムの暗号化

データベース内のカラムを暗号化する場合は、ENCRYPT 関数を使用します。ENCRYPT 関数は、同じ AES の強力な暗号化アルゴリズムを使用します。このアルゴリズムはデータベースの暗号化用に使用され、その関数に渡される値を暗号化します。

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

**警告**
強力な暗号化が適用されたデータベースの場合、キーのコピーは必ず安全な場所に保管してください。暗号化キーがわからなくなった場合は、Sybase 製品の保守契約を結んでいるサポートセンタに依頼してもデータにはアクセスできません。アクセスできなくなったデータベースは、廃棄して、新しくデータベースを作成する必要があります。

暗号化された値は、DECRYPT 関数で復号化できます。このとき、ENCRYPT 関数で指定したキーと同じキーを使用する必要があります。これらの関数はともに LONG BINARY 値を返します。異なるデータ型を使用する必要がある場合は、CAST 関数を使用して、その値を必要なデー

タ型に変換できます。次の例では、CAST 関数を使用して、復号化された値を必要なデータ型に変換する方法を示します。「CAST 関数 [データ型変換]」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

データベースユーザーが復号化された形式のデータにアクセスする必要があっても、暗号化キーにはアクセスできないようにする必要がある場合は、DECRYPT 関数を使用するビューを作成できます。これにより、ユーザーは暗号化キーを知らなくても、復号化されたデータにアクセスできるようになります。テーブルを使用したビューまたはストアドプロシージャーを作成する場合は、ALTER VIEW 文や ALTER PROCEDURES 文の SET HIDDEN パラメーターを使用して、ユーザーがビュー定義やプロシージャー定義を参照することによって暗号化キーにアクセスできないようにすることができます。「ALTER PROCEDURE 文」『SQL Anywhere サーバー SQL リファレンス』と「ALTER VIEW 文」『SQL Anywhere サーバー SQL リファレンス』を参照してください。

### 参照

- 「ENCRYPT 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』
- 「DECRYPT 関数 [文字列]」『SQL Anywhere サーバー SQL リファレンス』

### カラムの暗号化の例

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

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

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

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

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

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

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

データベースに新しいユーザーを追加する場合

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

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

このユーザーがパスワードを変更した場合

```
UPDATE user_info
SET user_pwd='xyz'
WHERE employee_ID='1';
```

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

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

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

## テーブル暗号化

テーブル暗号化によって、データベース全体の暗号化がもたらすようなパフォーマンスの低下を招くことなく、機密データが含まれるテーブルやマテリアライズドビューを暗号化することができます。テーブルの暗号化が有効な場合、暗号化されたテーブルのテーブルページ、関連するインデックスページ、テンポラリファイルのページが暗号化されます。暗号化されたテーブルのトランザクションを含むトランザクションログのページも暗号化されます。

マテリアライズドビューの暗号化については、「マテリアライズドビューの暗号化または復号化」『SQL Anywhere サーバー SQL の使用法』を参照してください。

データベース内のテーブルを暗号化するためには、テーブル暗号化を有効にしておく必要があります。テーブル暗号化の有効化は、データベースを初期化するときに行います。テーブル暗号化が有効になっているかどうかを確認するには、次のように DB_PROPERTY 関数を使用して EncryptionScope データベースプロパティの値を取得します。

```
SELECT DB_PROPERTY( 'EncryptionScope' );
```

TABLE が返された場合は、テーブル暗号化が有効になっています。

テーブル暗号化で暗号化アルゴリズムが有効であるかどうかを確認するには、次のように DB_PROPERTY 関数を使用して Encryption データベースプロパティの値を取得します。

```
SELECT DB_PROPERTY( 'Encryption' );
```

サポートされている暗号化アルゴリズムのリストについては、「データベースの暗号化と復号化」1396 ページを参照してください。

### テーブル暗号化がパフォーマンスに及ぼす影響

暗号化されたテーブルでは、各テーブルページがディスクへの書き込みと同時に暗号化され、ディスクから読み取るときに復号化されます。このプロセスはアプリケーションには影響しません。ただし、暗号化されたテーブルの読み込みや書き込みにおいてパフォーマンスが多少低下することがあります。既存のテーブルを暗号化または復号化する場合、テーブルのサイズによっては時間がかかることがあります。

暗号化されたテーブル内のカラムに対するインデックスのインデックスページ、暗号化されたテーブルのトランザクションを含むトランザクションログのページ、データベースのテンポラリファイルのすべてのページも暗号化されます。その他のデータベースとトランザクションログページは暗号化されません。

暗号化されたテーブルに圧縮されたカラムが含まれている場合があります。その場合、データは圧縮されてから暗号化されます。

テーブルの暗号化は必要記憶域には影響しません。

### テーブル暗号化が有効であるデータベースの起動

テーブル暗号化が有効であるデータベースを起動する方法は、暗号化されたデータベースを起動する場合と同じです。たとえば、-ek オプションを指定してデータベースを起動する場合は、キーを指定する必要があります。-ep オプションを指定してデータベースを起動すると、キーの入力を要求されます。「初期化ユーティリティ (dbinit)」958 ページを参照してください。

### データベース内のテーブル暗号化の有効化

テーブル暗号化を有効にする場合は、データベースの作成時に行う必要があります。データベースでテーブル暗号化が有効になっていない場合、またはデータベース暗号化が有効な場合は、テーブル暗号化を有効にしてデータベースを再作成する必要があります。

**♦ テーブル暗号化を有効にしてデータベースを作成するには、次の手順に従います (SQL の場合)。**

- CREATE DATABASE 文を使用してデータベースを作成します。このとき、キーと暗号化アルゴリズムを指定します。

  次のコマンドは、キー abc と暗号化アルゴリズム AES256_FIPS を使用して、強力なテーブル暗号化が有効になっているデータベース *new.db* を作成します。

  ```
  CREATE DATABASE 'new.db'
    ENCRYPTED TABLE
    KEY 'abc'
    ALGORITHM 'AES256_FIPS';
  ```

  このデータベース内のテーブルを暗号化するときには、AES256_FIPS アルゴリズムと abc キーが使用されます。

**♦ テーブル暗号化を有効にしてデータベースを作成するには、次の手順に従います (コマンドラインの場合)。**

- dbinit に -et オプションと -ek オプションを指定し、キーと暗号化アルゴリズムも指定して、データベースを作成します。

次のコマンドは、キー abc と暗号化アルゴリズム AES256_FIPS を使用して、強力なテーブル暗号化が有効になっているデータベース *new.db* を作成します。

```
dbinit new.db -et -ek abc -ea AES256_FIPS
```

このデータベース内のテーブルを暗号化するときには、AES256_FIPS アルゴリズムと abc キーが使用されます。

**♦ 既存のデータベースを使用して、テーブル暗号化を有効にしてデータベースを作成するには、次の手順に従います (SQL の場合)。**

● キーを指定して CREATE ENCRYPTED TABLE DATABASE 文を使用して、データベースの暗号化されたコピーを作成します。

次の例は、contacts1 という既存のデータベースから contacts2 というデータベースを作成します。新しいデータベースでは、テーブル暗号化がサポートされています。

```
CREATE ENCRYPTED TABLE DATABASE 'contacts2.db'
  FROM 'contacts1.db'
  KEY 'Sd8f6654'
  OLD KEY 'Sc8e5543';
```

このデータベース内のテーブルを暗号化するときには、AES アルゴリズムと Sd8f6654 キーが使用されます。

## テーブルの暗号化

データベース内のテーブルを暗号化するためには、そのデータベースでテーブル暗号化が有効になっている必要があります。「データベース内のテーブル暗号化の有効化」1406 ページを参照してください。

テーブルを暗号化するときは、データベースの作成時に指定した暗号化のアルゴリズムとキーが使用されます。

**♦ テーブルの作成時にテーブルを暗号化するには、次の手順に従います (SQL の場合)。**

● CREATE TABLE 文の ENCRYPTED 句を使用してテーブルを作成します。

次のコマンドは、暗号化されたテーブル MyEmployees を作成します。

```
CREATE TABLE MyEmployees (
 MemberID CHAR(40),
 CardNumber INTEGER )
ENCRYPTED;
```

**♦ 作成済みのテーブルを暗号化するには、次の手順に従います (SQL の場合)。**

● ALTER TABLE 文の ENCRYPTED 句を使用してテーブルを暗号化します。

次の文は、MyEmployees2 というテーブルを作成してから暗号化します。

```
CREATE TABLE MyEmployees2 (
  MemberID CHAR(40),
  CardNumber INTEGER );
```

```
ALTER TABLE MyEmployees2
  ENCRYPTED;
```

**参照**

- 「データベースの暗号化と復号化」1396 ページ
- 「初期化ユーティリティ (dbinit)」958 ページ
- 「暗号化されているデータベースの作成」1398 ページ
- 「データベースの作成 (Sybase Central)」118 ページ
- 「ALTER TABLE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「CREATE DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「CREATE ENCRYPTED DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「CREATE DECRYPTED DATABASE 文」『SQL Anywhere サーバー SQL リファレンス』
- 「CREATE TABLE 文」『SQL Anywhere サーバー SQL リファレンス』

# Windows Mobile データベースセキュリティ

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

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

Windows Mobile 上で動作するデータベースは、Windows デスクトッププラットフォームで動作するデータベースと同じユーザー識別情報と認証機能を使用します。これらの機能により、データベースにアクセスできるユーザーと、そのユーザーが実行できるアクションが制御されます。「データベースアクセス」1382 ページを参照してください。

## Windows Mobile デバイスセキュリティ

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

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

## データベースサーバーオプション

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

このオプションは、Windows Mobile デバイス上でデータベースを起動するときに、**[サーバー起動オプション]** ウィンドウの [オプション] フィールドで設定します。

詳細については、「コマンドラインからパーミッションを制御する」159 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Windows Mobile でのオプションの設定については、「Windows Mobile のサーバーオプション」427 ページを参照してください。

## 監査

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

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

詳細については、「データベースアクティビティの監査」1388 ページを参照してください。

## Windows Mobile 上でのデータベースの暗号化

データベース暗号化機能を使用する際の、暗号化のレベルを選択します。データベースを安全に管理するために、単純暗号化または強力な暗号化のいずれかを選択できます。SQL Anywhere は、Windows Mobile 上で、単純暗号化と強力な暗号化の両方をサポートしています。

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

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

**強力な暗号化**　このレベルの暗号化は、データベースやトランザクションログファイルに含まれる情報を難読化することで、ディスクユーティリティを使用してファイルを表示するだけではデータを解読できないようにします。強力な暗号化にすると、キーなしではデータベースにまったくアクセスできなくなります。Windows Mobile では、ARM プロセッサー用に AES_FIPS および AES256_FIPS アルゴリズムのみがサポートされています。

詳細については、「データベースの暗号化と復号化」1396 ページを参照してください。

## 通信の暗号化と Windows Mobile

クライアント／サーバー通信を暗号化して、ネットワーク上の通信のセキュリティを強化できます。SQL Anywhere は、単純暗号化と強力な暗号化の 2 種類の通信暗号化を備えています。

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

強力な通信暗号化は、Windows Mobile では利用できません。

通信の暗号化の詳細については、「Encryption (ENC) 接続パラメーター」342 ページを参照してください。

# トランスポートレイヤーセキュリティ

**注意**
別途ライセンスが必要な必須コンポーネント

ECC 暗号化と FIPS 認定の暗号化には、別途ライセンスが必要です。強力な暗号化テクノロジはすべて、輸出規制対象品目です。

「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

トランスポートレイヤーセキュリティは IETF 標準プロトコルであり、デジタル証明書とパブリックキー暗号方式を使用して、クライアント／サーバー通信をセキュリティ保護します。トランスポートレイヤーセキュリティにより、暗号化、改ざん検出、証明書ベースの認証が実現します。

トランスポートレイヤーセキュリティは、次の場合に使用できます。

- SQL Anywhere データベースサーバーとクライアントアプリケーションとの間の通信のセキュリティ保護。
- Mobile Link サーバーと Mobile Link クライアントとの間の通信のセキュリティ保護。
- セキュアな SQL Anywhere Web サーバーの設定。

セキュリティ保護された通信は、次のメッセージの交換 (ハンドシェイク) で始まります。

- **サーバー認証**　トランスポートレイヤーセキュリティでは、セキュア接続の確立と保護にサーバー証明書を使用します。サーバーごとにユニークな証明書ファイルを作成します。SQL Anywhere クライアント／サーバー通信用または Mobile Link 同期用のサーバー認証を使用できます。
  - SQL Anywhere クライアント／サーバー通信では、データベースクライアントは SQL Anywhere データベースサーバーの ID を検証します。
  - Mobile Link 同期では、Mobile Link クライアント (SQL Anywhere または Ultra Light) は Mobile Link サーバーの ID を検証します。

## 効率

トランスポートレイヤーセキュリティプロトコルでは、パブリックキー暗号化と対称キー暗号化を組み合わせて使用します。パブリックキー暗号化は、認証テクニックとして優れていますが、処理量が多くなります。セキュア接続が確立されると、クライアントとサーバーはそれ以降の通信に、キーサイズが 128 ビットで高効率の対称暗号を使用します。

## 証明書

SQL Anywhere には createcert というツールが付属しています。このツールを使用すると、トランスポートレイヤーセキュリティ用の X.509 証明書ファイルを作成できます。一方、サードパーティ証明書の存在を確認する必要がある場合、またはセキュリティのより高い証明書が必要な場合は、証明書を認証局から購入してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### データベースファイルの暗号化

データベースファイルの暗号化の詳細については、次を参照してください。

- SQL Anywhere データベース：「データベースの暗号化と復号化」
- Ultra Light データベース：「Ultra Light データベースのセキュリティ」『Ultra Light データベース管理とリファレンス』

## TLS サポート

> **注意**
> 別途ライセンスが必要な必須コンポーネント
>
> ECC 暗号化と FIPS 認定の暗号化には、別途ライセンスが必要です。強力な暗号化テクノロジはすべて、輸出規制対象品目です。
>
> 「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

このトピックでは、RSA、ECC、FIPS 認定の各暗号化のサポートについて詳しく説明します。

### RSA 暗号化

RSA 暗号化は SQL Anywhere では無料で提供されており、クライアント／サーバー通信、同期、Web サービスに使用できます。無料バージョンは FIPS 認定ではありません。FIPS 認定の RSA 暗号化を実装するには、別途ライセンスが必要です。

RSA がサポートされているプラットフォームのリストについては、http://www.ianywhere.jp/tech/1061806-os_components.html を参照してください。

### ECC 暗号化

ECC 暗号化を実装するには、別途ライセンスが必要です。

ECC がサポートされているプラットフォームのリストについては、http://www.ianywhere.jp/tech/1061806-os_components.html を参照してください。

### FIPS 認定の暗号化

FIPS 認定の暗号化は、RSA 暗号化でのみ利用可能です。(ECC は FIPS プログラムで未対応)。

FIPS 認定の暗号化には別のライセンスが必要です。「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

FIPS 認定コンポーネントがサポートされているプラットフォームのリストについては、http://www.ianywhere.jp/tech/1061806-os_components.html を参照してください。

# FIPS 認定の暗号化テクノロジ

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

連邦情報処理規格 (FIPS) 140-2 では、セキュリティアルゴリズムの要件を指定しています。FIPS 140-2 は、米国商務省標準技術局 (NIST : National Institute of Standards and Technology) およびカナダ通信安全保障局 (CSE：Canadian Communications Security Establishment) を通じて、米国政府とカナダ政府から付与されます。

SQL Anywhere では、2 つの FIPS 認定モジュールが使用されており、どちらも Certicom 製です。Windows (デスクトップと Windows Mobile) と UNIX プラットフォームの場合、SQL Anywhere は Certicom Security Builder GSE (FIPS Module v2.0) を使用します。これは次のページにある 542 番です。

### FIPS の強制

必要に応じて、FIPS オプションを使用して、クライアントまたはサーバーでの FIPS 認定暗号化の使用を強制できます。FIPS オプションをオンに設定すると、セキュリティ保護された通信はすべて FIPS 認定である必要があります。ユーザーが非 FIPS の RSA 暗号化を使用しようとした場合、その RSA は自動的に FIPS 認定 RSA にアップグレードされます。ECC を選択した場合は、エラーがレポートされます (ECC は FIPS 認定暗号化をサポートしていません)。FIPS オプションは、FIPS 認定暗号化を強制的に使用するクライアントまたはサーバーで設定できます。SQL Anywhere サーバーと Mobile Link サーバーには -fips コマンドラインオプションが用意されています。クライアントには fips オプションがあり、暗号化接続パラメーターを使用して設定できます。

SQL Anywhere データベースファイルの FIPS 認定暗号化を使用した暗号化の詳細については、「強力な暗号化」1397 ページを参照してください。

# トランスポートレイヤーセキュリティの設定

次の手順には、トランスポートレイヤーセキュリティの設定に必要なタスクの概要が記載されています。

1. デジタル証明書を取得します。

   ID ファイルと証明書ファイルが必要です。サーバー ID ファイルにはサーバーのプライベートキーが含まれているので、データベースサーバーまたは Mobile Link サーバーにセキュリティ保護された状態で格納する必要があります。サーバー証明書ファイルはクライアントに配布します。

   証明書は認証局から購入することができます。SQL Anywhere には、証明書を作成する機能もあり、特に開発やテストのときに便利です。「デジタル証明書」1414 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

2. SQL Anywhere クライアント／サーバーアプリケーション用のトランスポートレイヤーセキュリティを設定する場合は、次の手順に従います。

   - **トランスポートレイヤーセキュリティを指定して SQL Anywhere データベースサーバーを起動する**　-ec データベースサーバーオプションを使用して、セキュリティのタイプ、サーバー ID ファイル名、サーバーのプライベートキーを保護するパスワードを指定します。

     共有メモリを経由した暗号化されていない接続も許可する場合は、-es オプションを指定します。

     「トランスポートレイヤーセキュリティを使用するデータベースサーバー」1420 ページを参照してください。

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

     「トランスポートレイヤーセキュリティを使用するようにクライアントアプリケーションを設定する」1422 ページを参照してください。

3. SQL Anywhere Web サービス用のトランスポートレイヤーセキュリティを設定する場合は、次の手順に従います。

   - **トランスポートレイヤーセキュリティを指定して SQL Anywhere データベースサーバーを起動する**　-xs データベースサーバーオプションを使用して、セキュリティのタイプ、サーバー ID ファイル名、サーバーのプライベートキーを保護するパスワードを指定します。

   - **ブラウザーまたは他の Web クライアントが証明書を信頼するように設定する**　「SQL Anywhere Web サービスの暗号化」1425 ページを参照してください。

4. Mobile Link 同期用のトランスポートレイヤーセキュリティを設定する場合は、次の手順に従います。

   - **トランスポートレイヤーセキュリティを指定して Mobile Link サーバーを起動する**　mlsrv12 -x オプションを使用して、セキュリティストリーム、サーバー ID ファイル名、サーバーのプライベートキーを保護するパスワードを指定します。

     「トランスポートレイヤーセキュリティを使用する Mobile Link サーバーの起動」1427 ページを参照してください。

   - **トランスポートレイヤーセキュリティを使用するように Mobile Link クライアントを設定する**　Mobile Link 同期クライアントユーティリティ (dbmlsync) または Ultra Light アプリケーションを使用して、適切なセキュリティまたはネットワークプロトコルオプションを指定します。セキュリティストリームと信頼できるサーバー証明書ファイル名を指定します。

     「トランスポートレイヤーセキュリティを使用する MobiLink クライアントの設定」1429 ページを参照してください。

# デジタル証明書

トランスポートレイヤーセキュリティを設定するには、デジタル証明書が必要です。証明書は、認証局から取得することも、SQL Anywhere の機能を使用して作成することもできます。

### SQL Anywhere 証明書作成ユーティリティ

SQL Anywhere 証明書作成ユーティリティ createcert は、RSA または ECC を使用して X.509 証明書ファイルを生成します。「証明書作成ユーティリティ (createcert)」924 ページを参照してください。

### SQL Anywhere 証明書ビューアーユーティリティ

SQL Anywhere 証明書ビューアーユーティリティ viewcert は、RSA または ECC を使用して X.509 証明書を読み込みます。「証明書ビューアーユーティリティ (viewcert)」934 ページを参照してください。

### サーバー認証に使用する証明書

サーバー認証に使用する証明書ファイルは、同じ手順で作成できます。どちらの場合も、ID ファイルと証明書ファイルを作成します。

サーバー認証の場合は、サーバー ID ファイルとクライアントに配布する証明書ファイルを作成します。「エンタープライズルート証明書」1417 ページを参照してください。

### 証明書設定

証明書には、自己署名されたものと、民間またはエンタープライズ認証局によって署名されたものがあります。

- **自己署名証明書**　自己署名されたサーバー証明書は、単純な設定で使用します。「自己署名ルート証明書」1415 ページを参照してください。

- **エンタープライズルート証明書**　エンタープライズルート証明書を使用すると、サーバー証明書に署名できるため、複数のサーバーが配備されている環境でのデータの整合性と拡張性が向上します。

  - サーバー証明書の署名に使用するプライベートキーは、安全な中央のロケーションに保存できます。
  - サーバー認証では、クライアントを再設定しなくても Mobile Link サーバーまたはデータベースサーバーを追加できます。

  「証明書チェーン」1415 ページを参照してください。

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

  「証明書チェーン」1415 ページと「グローバル署名証明書」1417 ページを参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## 自己署名ルート証明書

自己署名ルート証明書は、単一の Mobile Link サーバーまたはデータベースサーバーで構成される単純な設定において使用されます。

**ヒント**
サーバー ID ファイルが複数必要な場合は、エンタープライズレベルの証明書チェーンまたは民間認証局を使用します。認証局にはルートプライベートキーを格納する専用の設備があり、拡張性と証明書の高度な整合性を提供します。

「証明書チェーン」1415 ページを参照してください。

- **証明書**　サーバー認証証明書の場合、自己署名証明書はクライアントに配布されます。自己署名証明書は、識別情報、サーバーのパブリックキー、自己署名されたデジタル署名を含む電子文書です。

- **ID ファイル**　サーバー認証証明書の場合、ID ファイルはセキュリティ保護された状態で Mobile Link サーバーまたはデータベースサーバーに格納されます。ID ファイルは、自己署名証明書 (クライアントに配布) と、対応するプライベートキーを組み合わせたものです。プライベートキーがあると、Mobile Link サーバーまたはデータベースサーバーは、初期ハンドシェイクでクライアントから送信されたメッセージを復号できます。

**参照**


- 「サーバー認証」1429 ページ
- 「トランスポートレイヤーセキュリティを使用するデータベースサーバー」1420 ページ
- 「証明書作成ユーティリティ (createcert)」924 ページ


## 証明書チェーン

ID ファイルが複数必要な場合、自己署名証明書の代わりに証明書チェーンを使用することで、セキュリティと拡張性を向上させることができます。証明書チェーンでは、ID の署名に認証局またはエンタープライズルート証明書が必要です。

「自己署名ルート証明書」1415 ページを参照してください。

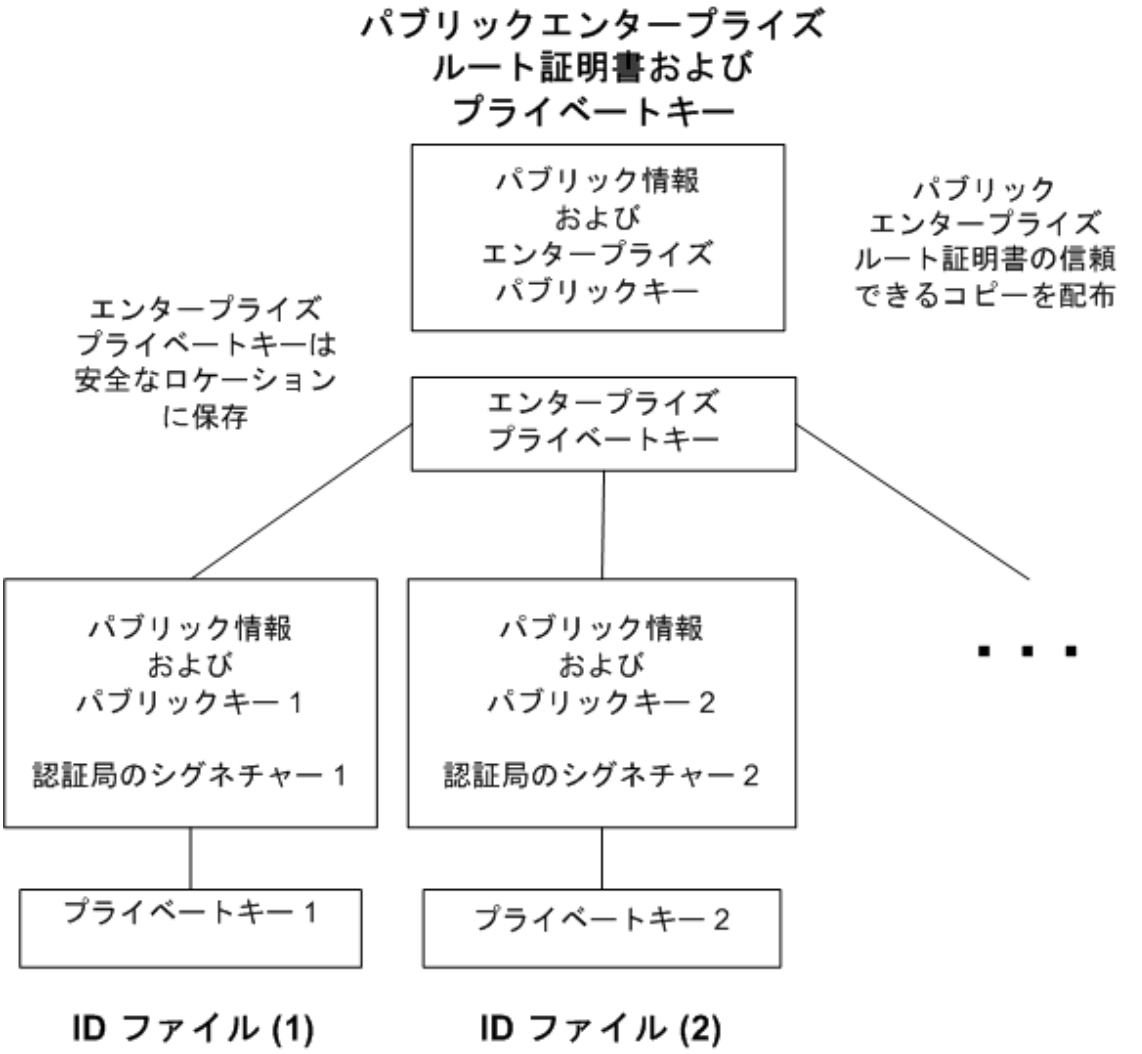
### 証明書チェーンを使用する利点

証明書チェーンには、次の利点があります。

- **拡張性**　サーバー認証の場合、エンタープライズルート証明書または認証局によって署名されたすべての証明書を信頼するようにクライアントを設定できます。新しい Mobile Link サーバーまたはデータベースサーバーを追加する場合、クライアントに新しい証明書のコピーは不要です。

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

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




**マルチサーバー環境での証明書の使用**

マルチサーバー環境で使用する証明書を作成するには、次の手順に従います。

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

  エンタープライズプライベートキーは安全なロケーションに保存します。専用の設備の方が安全です。

  サーバー認証の場合、パブリックエンタープライズルート証明書をクライアントに配布します。

- エンタープライズルート証明書を使用して、ID に署名します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

パブリックエンタープライズルート証明書とエンタープライズプライベートキーを使用して、各 ID に署名します。サーバー認証の場合、ID ファイルはサーバー用に使用します。

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

**参照**


- 「証明書作成ユーティリティ (createcert)」924 ページ
- 「グローバル署名証明書」1417 ページ


## エンタープライズルート証明書

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

- 信頼できる証明書を作成するために使用されるプライベートキーは、専用の設備に保存できます。
- サーバー認証では、クライアントを再設定しなくてもサーバーを追加できます。

エンタープライズルート証明書を設定するには、エンタープライズルート証明書と、ID の署名に使用するエンタープライズプライベートキーを作成します。

サーバー証明書の作成の詳細については、「署名付き ID ファイル」1417 ページを参照してください。

エンタープライズルート証明書の生成の詳細については、「グローバル署名証明書」1417 ページを参照してください。

## 署名付き ID ファイル

エンタープライズルート証明書を使用して、サーバーの ID ファイルに署名できます。

サーバー認証の場合、各サーバー用に ID ファイルを生成します。これらの証明書はエンタープライズルート証明書によって署名されるので、createcert -s オプションを使用します。

署名付き ID ファイルの生成の詳細については、「証明書作成ユーティリティ (createcert)」924 ページを参照してください。

# グローバル署名証明書

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

グローバル署名証明書には、次の利点があります。

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

### グローバル署名証明書の設定

グローバル署名 ID ファイルを設定するには、次の手順に従います。

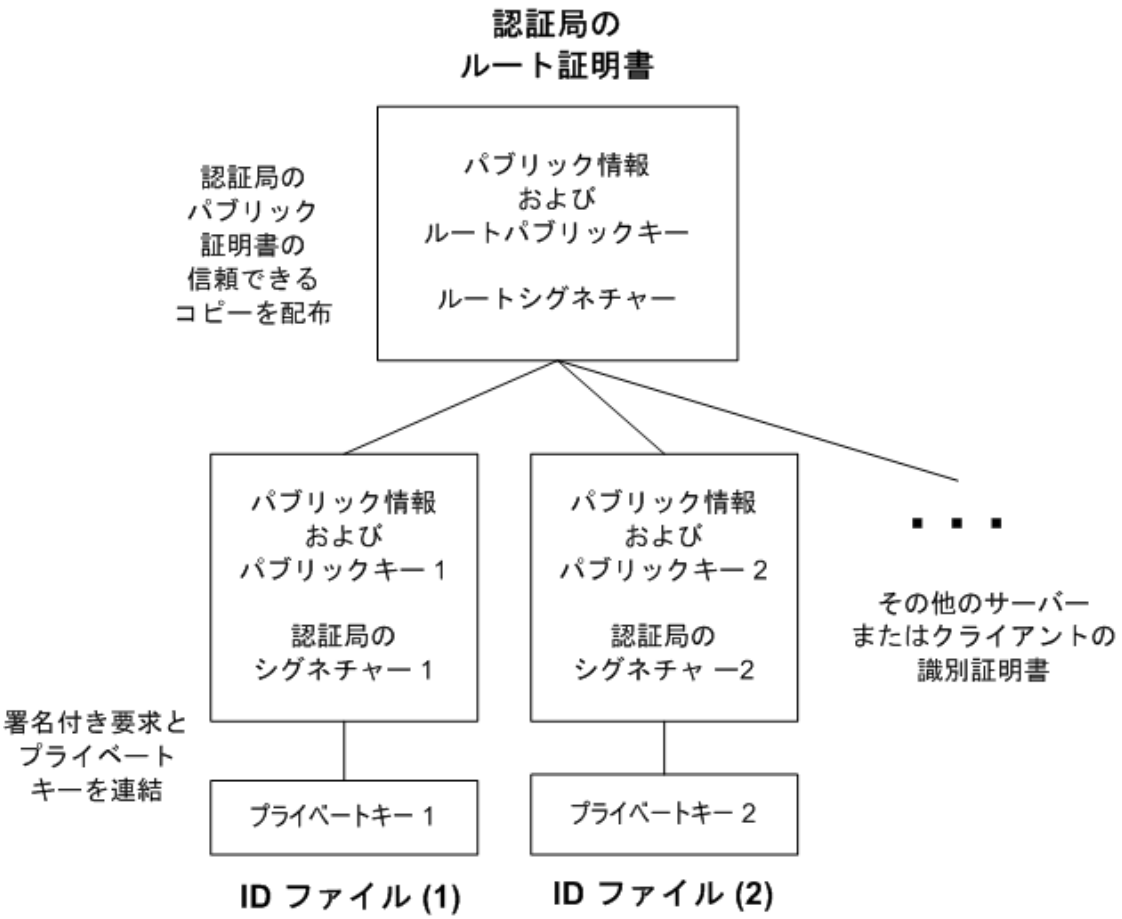
- -r オプションを指定した createcert ユーティリティを使用して証明書を要求します。「証明書作成ユーティリティ (createcert)」924 ページを参照してください。
- 認証局を使用して各要求に署名します。署名付き要求と対応するプライベートキーを組み合わせて、サーバーの ID ファイルを作成できます。

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

## グローバル署名 ID ファイル

グローバル署名証明書を直接サーバーの ID ファイルとして使用できます。次の図は、複数の ID ファイルの設定を示します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1




dbsrv12 または mlsrv12 コマンドラインで、サーバー ID ファイルと、プライベートキーのパスワードを参照します。

**参照**

- SQL Anywhere :「トランスポートレイヤーセキュリティを使用するデータベースサーバー」1420 ページ
- Mobile Link :「トランスポートレイヤーセキュリティを使用する Mobile Link サーバーの起動」1427 ページ

## 認証局の証明書を信用するようにクライアントを設定する

サーバー認証の場合は、サーバーにアクセスするクライアントがチェーン内のルート証明書を信頼することを確認する必要があります。グローバル署名証明書の場合、ルート証明書は認証局の証明書です。

**注意**
グローバル署名証明書を使用する場合、各クライアントはフィールド値を確認して、同じ認証局が他のクライアント用に署名した証明書を信頼することを避けなければなりません。

「証明書フィールドの確認」1430 ページを参照してください。

サーバー証明書を信頼するように Mobile Link クライアントを設定する方法については、「トランスポートレイヤーセキュリティを使用する MobiLink クライアントの設定」1429 ページを参照してください。

トランスポートレイヤーセキュリティを使用するデータベースサーバーを構成する方法の詳細については、「トランスポートレイヤーセキュリティを使用するデータベースサーバー」1420 ページを参照してください。

グローバル署名証明書を使用して信頼性を確立する方法については、「グローバル署名証明書」1417 ページを参照してください。

# SQL Anywhere クライアント／サーバー通信の暗号化

SQL Anywhere のクライアント／サーバー通信は、トランスポートレイヤーセキュリティを使用して暗号化できます。

**参照**

- 「SQL Anywhere Web サービスの暗号化」1425 ページ

## トランスポートレイヤーセキュリティを使用するデータベースサーバー

トランスポートレイヤーセキュリティを使用してデータベースサーバーを起動するには、サーバー ID ファイル名と、サーバーのプライベートキーを保護するパスワードを指定します。

トランスポートレイヤーセキュリティを設定する手順の概要については、「トランスポートレイヤーセキュリティの設定」1412 ページを参照してください。

-ec データベースサーバーオプションを使用して、identity パラメーターと identity_password パラメーターを指定します。共有メモリを経由した暗号化されていない接続を許可する場合は、-es オプションも指定する必要があります。

dbsrv12 コマンドラインの一部の構文を次に示します。

```
-ec tls(
  tls_type=algorithm;
  identity=server-identity-filename;
  identity_password=password )
-x tcpip
```

- ***algorithm*** 使用するアルゴリズムアルゴリズムの場合は **rsa** を指定し、RSA および ECC 暗号化の場合は **ecc** を指定します。FIPS 認定の RSA 暗号化の場合は、**tls_type=rsa;fips=y** を指

定します。RSA FIPS 認定暗号化は別の認定ライブラリを使用しますが、SQL Anywhere 9.0.2 以降で RSA を使用しているクライアントと互換性があります。

FIPS 認定コンポーネントがサポートされているプラットフォームのリストについては、http://www.ianywhere.jp/tech/1061806-os_components.html を参照してください。

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

FIPS 認定のアルゴリズムの実行については、「-fips dbeng12/dbsrv12 サーバーオプション」230 ページを参照してください。

- ***server-identity-filename*** サーバー ID ファイルのパスとファイル名を指定します。FIPS 認定の RSA 暗号化を使用している場合は、RSA アルゴリズムを使用して証明書を生成する必要があります。

  ID ファイルには、パブリック証明書とプライベートキーが格納されています。自己署名されていない証明書の場合は、その証明書に署名を行うすべての証明書も ID ファイルに格納されています。

  サーバー証明書の作成については、「デジタル証明書」1414 ページを参照してください。サーバー証明書は、自己署名証明書、または認証局やエンタープライズルート証明書の署名を受けた証明書のいずれかです。

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

単純暗号化を使用してデータベースサーバーを起動することもできます。単純暗号化を使用すると、パケットスニッファを使用して、クライアントとサーバーの間で送信されるネットワークパケットを読み取るのが困難になります。ただし、データの整合性は保証されず、サーバー認証を行うこともできません。

TCP/IP プロトコルは、-x データベースサーバーオプションを使用して指定します。

## 参照


- 「-ec dbeng12/dbsrv12 サーバーオプション」223 ページ
- 「-es dbeng12/dbsrv12 サーバーオプション」227 ページ
- 「-x dbeng12/dbsrv12 サーバーオプション」289 ページ


## 例

次の例 (すべて 1 行に入力) では、-ec データベースサーバーオプションを使用して、ECC セキュリティ、サーバー ID ファイル、サーバーのプライベートキーを保護するパスワードを指定します。

```
dbsrv12 -ec tls( tls_type=ecc;identity=c:¥test¥serv1_ecc.id;identity_password=mypwd )
 -x tcpip c:¥test¥secure.db
```

設定ファイルとファイル難読化ユーティリティ (dbfhide) を使用して、パスワードを含むコマンドラインオプションを非表示にできます。「ファイル非表示ユーティリティ (dbfhide)」953 ページと「@data dbeng12/dbsrv12 サーバーオプション」201 ページを参照してください。

# トランスポートレイヤーセキュリティを使用するようにクライアントアプリケーションを設定する

トランスポートレイヤーセキュリティを使用するように、SQL Anywhere クライアントアプリケーションを設定できます。暗号化接続パラメーターセットを使用して、信頼できる証明書、暗号化のタイプ、ネットワークプロトコルを指定します。

トランスポートレイヤーセキュリティを設定する手順の概要については、「トランスポートレイヤーセキュリティの設定」1412 ページを参照してください。

## サーバー認証

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

### デジタル署名

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

●証明書で実行されるアルゴリズムが、ユニークな値またはハッシュを生成します。

●証明書への署名または認証局のプライベートキーを使用して、ハッシュが暗号化されます。

●デジタル署名と呼ばれる暗号化ハッシュが、証明書に埋め込まれます。

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

クライアントアプリケーションがデータベースサーバーにアクセスする場合、各クライアントがトランスポートレイヤーセキュリティを使用するように設定されていると、サーバーはその証明書のコピーをクライアントに送信します。クライアントは、証明書に含まれているサーバーのパブリックキーを使用して証明書のデジタル署名を復号化し、証明書の新しいハッシュを算出して、2 つの値を比較します。値が一致する場合は、サーバーの証明書の整合性が確認されます。

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

自己署名証明書の詳細については、「自己署名ルート証明書」1415 ページを参照してください。

エンタープライズルート証明書と認証局の詳細については、「証明書チェーン」1415 ページを参照してください。

### 証明書フィールドの確認

グローバル署名証明書を使用する場合、各クライアントは証明書のフィールド値を確認して、同じ認証局が他のクライアント用に署名した証明書を信頼することを避けなければなりません。この問題を解決するには、証明書の識別情報部分のフィールド値をテストするようにクライアン

トに要求します。認証局は、署名を行ったすべての証明書の識別情報が正確であることを保証する必要があります。

グローバル署名証明書の詳細については、「グローバル署名証明書」1417 ページを参照してください。

createcert ユーティリティを使用して証明書を作成する場合は、組織、組織単位、通称のフィールドに値を入力します。対応するクライアント接続パラメーターを使用して、これらのフィールドを確認します。サードパーティの認証局を使用して証明書にグローバル署名している場合は、証明書フィールドを確認してください。

- **組織**　組織フィールドは、certificate_company 暗号化プロトコルオプションに対応します。「certificate_company プロトコルオプション」374 ページを参照してください。
- **組織単位**　組織単位フィールドは、certificate_unit 暗号化プロトコルオプションに対応します。「certificate_unit プロトコルオプション」376 ページを参照してください。
- **通称**　通称フィールドは、certificate_name 暗号化プロトコルオプションに対応します。「certificate_name プロトコルオプション」375 ページを参照してください。

クライアント側の暗号化接続パラメーターの詳細については、「Encryption (ENC) 接続パラメーター」342 ページを参照してください。

### trusted_certificate プロトコルオプションの使用

Encryption 接続パラメーターで TLS を指定する場合、これが唯一の必須プロトコルオプションです。クライアントは、trusted_certificate 暗号化プロトコルオプションを使用して、信用されたデータベースサーバー証明書を指定します。信頼できる証明書は、サーバーの自己署名証明書、パブリックエンタープライズルート証明書、民間認証局に属する証明書のいずれかです。

### 参照

- 「trusted_certificate プロトコルオプション」401 ページ
- 「デジタル証明書」1414 ページ

## トランスポートレイヤーセキュリティを使用するクライアント接続の確立

クライアントアプリケーションがトランスポートレイヤーセキュリティを使用するように設定するには、接続文字列の中で Encryption (ENC) 接続パラメーターを使用します。接続文字列の形式は次のとおりです (全体を 1 行で入力してください)。

```
Encryption=tls(
  tls_type=algorithm;
  [ fips={ y | n }; ]
  trusted_certificate=public-certificate
[ certificate_company=organization; ]
[ certificate_name=common-name; ]
[ certificate_unit=organization-unit ] )
```

- ***algorithm***　RSA 暗号化の場合は **rsa** を指定し、ECC 暗号化の場合は **ecc** を指定します。デフォルトは **rsa** です。FIPS 認定の RSA 暗号化の場合は、**tls_type=rsa;fips=y** を指定します。

RSA FIPS 認定暗号化は別の認定ライブラリを使用しますが、SQL Anywhere 9.0.2 以降で RSA を使用しているデータベースサーバーと互換性があります。**fips=y** と **tls_type=ecc** を同時に指定することはできません。

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

- ***public-certificate*** 信頼できる証明書を 1 つ以上含むファイルのパスとファイル名を指定します。FIPS 認定の RSA 暗号化を使用している場合は、RSA を使用して証明書を生成する必要があります。「trusted_certificate プロトコルオプション」401 ページを参照してください。

- ***organization*** 証明書に記されている組織フィールドがこの値と一致する場合にだけ、クライアントでサーバー証明書を受け入れるようにします。「certificate_company プロトコルオプション」374 ページを参照してください。

- ***common-name*** 証明書に記されている通称フィールドがこの値と一致する場合にだけ、クライアントでサーバー証明書を受け入れるようにします。「certificate_name プロトコルオプション」375 ページを参照してください。

- ***organization-unit*** 証明書に記されている組織単位フィールドがこの値と一致する場合にだけ、クライアントでサーバー証明書を受け入れるようにします。「certificate_unit プロトコルオプション」376 ページを参照してください。

trusted_certificate および他のクライアントセキュリティパラメーターの詳細については、「証明書フィールドの確認」1422 ページと「trusted_certificate プロトコルオプションの使用」1423 ページを参照してください。

## 参照


- 「デジタル証明書」1414 ページ
- 「Encryption (ENC) 接続パラメーター」342 ページ


## 例

次の例では、trusted_certificate 暗号化接続パラメーターを使用して、証明書 *public_cert.crt* を指定します。

```
"UID=DBA;PWD=sql;Host=myhost;Server=myserver;
ENC=tls(tls_type=ecc;trusted_certificate=public_cert.crt)"
```

次の例では、trusted_certificate 暗号化接続パラメーターを使用して証明書 *public_cert.crt* を指定し、certificate_unit および certificate_name 暗号化接続パラメーターを使用して証明書フィールドを確認します。

```
"UID=DBA;PWD=sql;Host=myhost;Server=myserver;
ENC=tls(tls_type=ecc;trusted_certificate=public_cert.crt;
certificate_unit=test_unit;certificate_name=my_certificate)"
```

# SQL Anywhere Web サービスの暗号化

SQL Anywhere Web サーバーは、SSL バージョン 3.0 と TLS バージョン 1.0 を使用した HTTPS 接続をサポートしています。

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

- **デジタル証明書を取得する**　データベースサーバー証明書ファイルと ID ファイルが必要です。証明書 (認証局の証明書の場合もあります) は、ブラウザーまたは Web クライアントに配布します。サーバー ID ファイルは、SQL Anywhere Web サーバーに安全に保存されます。

  認証局の使用に関する情報を含む、デジタル証明書の一般的な情報については、「デジタル証明書」1414 ページを参照してください。

- **トランスポートレイヤーセキュリティを指定して Web サーバーを起動する**　-xs データベースサーバーオプションを使用して、HTTPS、サーバー ID ファイル、プライベートキーを保護するパスワードを指定します。

  dbsrv12 コマンドラインの一部の構文を次に示します。

  **-xs** *protocol***(**
  **[ fips={ y | n }; ]**
  **identity=***server-identity-filename***;**
  **identity_password=***password***;... ) ...**

  - ***protocol***　**https** を指定するか、FIPS 認定の RSA 暗号化の場合は **fips=y** を付けて **https** を指定します。FIPS 認定の HTTPS は別の認定ライブラリを使用しますが、HTTPS と互換性があります。

    FIPS 認定のアルゴリズムの実行については、「-fips dbeng12/dbsrv12 サーバーオプション」230 ページを参照してください。

  - ***server-identity-filename***　サーバー ID のパスとファイル名を指定します。HTTPS では、RSA 証明書を使用する必要があります。

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

  -xs サーバーオプションの詳細については、「-xs dbeng12/dbsrv12 サーバーオプション」294 ページを参照してください。

  identity パラメーターと identity_password パラメーターの詳細については、次の項を参照してください。

  - 「Identity プロトコルオプション」383 ページ
  - 「Identity_Password プロトコルオプション」384 ページ

- **Web クライアントを設定する**　ブラウザーまたは他の Web クライアントが証明書を信頼するように設定します。信頼できる証明書は、自己署名証明書、エンタープライズルート証明書、または認証局証明書です。

認証局の使用に関する情報を含む、デジタル証明書の一般的な情報については、「デジタル証明書」1414 ページを参照してください。

# Mobile Link サーバー／クライアント通信の暗号化

Mobile Link のクライアント／サーバー通信は、トランスポートレイヤーセキュリティを使用して暗号化できます。

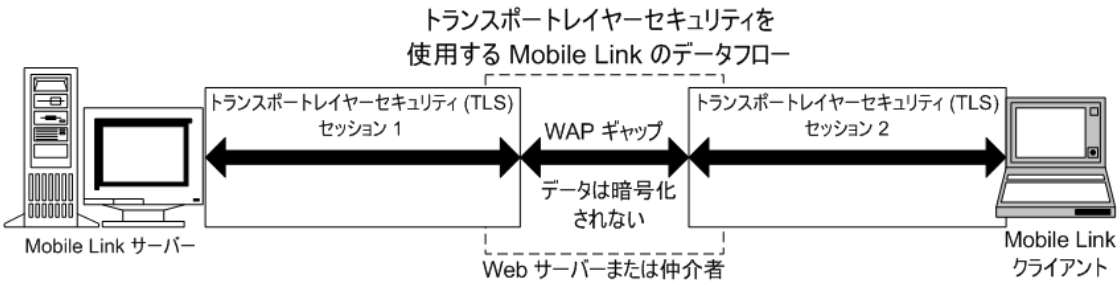
## エンドツーエンド暗号化

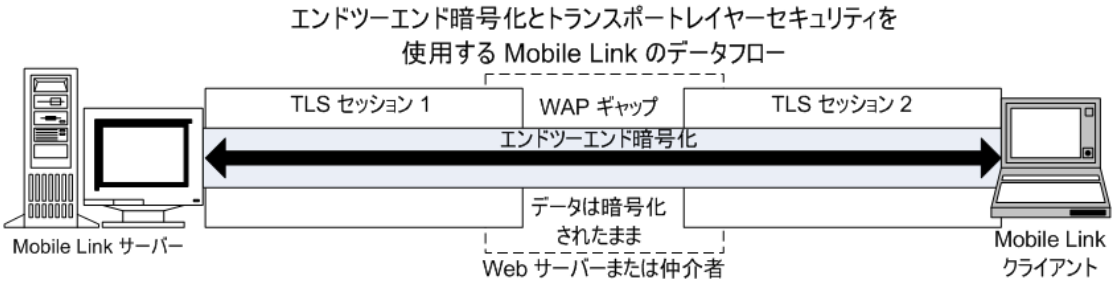
エンドツーエンド暗号化では、データは発信元で暗号化され、最終的な宛先で復号化されます。転送の途中もデータは常に暗号化されています。

Mobile Link TLS は、クライアントとサーバーの間の仲介者 (たとえば暗号化／復号化ハードウェア) までのデータを暗号化するためだけに使用されることもあります。仲介者に届いたデータは復号化され、その後の転送を行うために仲介者によって再び暗号化されます。特に、この動作は、Web サーバー経由で HTTPS を使用して同期するときに行われます。仲介者でデータの暗号化が解除される短い時間のことを、WAP (ワイヤレスアプリケーションプロトコル) ギャップと呼ぶことがあります。

企業内では、仲介者が企業の管理下にあるかぎり、多くの場合 WAP ギャップがあっても問題はありません。しかし、サードパーティにより実行される環境で、複数の企業からのデータが同じ WAP ギャップを通過する場合は、機密データが公開される可能性があります。エンドツーエンド暗号化では、同期ストリームが最初から最後まで暗号化されており、さらにもう一度 TLS を使用して暗号化することもできるため、仲介者によるデータへのアクセスを回避できます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1




MobiLink クライアントでエンドツーエンドの暗号化を有効にする方法については、次のトピックを参照してください。

- 「e2ee_type」『Mobile Link クライアント管理』
- 「e2ee_public_key」『Mobile Link クライアント管理』

MobiLink サーバーでエンドツーエンドの暗号化を設定する方法については、次のトピックを参照してください。

- 「-x mlsrv12 オプション」『Mobile Link サーバー管理』

**参照**

- 「キーペアジェネレーターユーティリティ (createkey)」981 ページ

## トランスポートレイヤーセキュリティを使用する Mobile Link サーバーの起動

トランスポートレイヤーセキュリティを使用して Mobile Link サーバーを起動するには、ID ファイルと、サーバーのプライベートキーを保護する ID パスワードを指定します。

トランスポートレイヤーセキュリティを設定する手順の概要については、「トランスポートレイヤーセキュリティの設定」1412 ページを参照してください。

### TCP/IP と HTTPS を使用する Mobile Link サーバーの保護

mlsrv12 -x サーバーオプションを使用して、ID と ID パスワードを指定します。次に示すのは mlsrv12 コマンドラインの一部です (全体を 1 行で入力する必要があります)。

**-x** *protocol*(
  **tls_type=***algorithm***;**
  **fips={ y | n };**
  **identity=***identity-file***;**
  **identity_password=***password***;...** )

- ***protocol*** 使用するプロトコルです。**https** または **tls** を指定します。**tls** で指定されるプロトコルは TLS を使用する TCP/IP です。

- ***algorithm*** 使用するアルゴリズム RSA 暗号化の場合は **rsa** を指定し、ECC 暗号化の場合は **ecc** を指定します。アルゴリズムは、ID ファイルを作成するときに使用される暗号化と一致する必要があります。

- **fips** FIPS 認定暗号化を使用するかどうかを示します。fips オプションは、RSA 暗号化のみと組み合わせて使用できます。RSA FIPS の場合は、Certicom の別の FIPS 140-2 認定ソフトウェアを使用します。fips オプションを使用するサーバーは、fips オプションを使用しないクライアントと互換性があります。逆についても同様です。RSA FIPS を使用できるのは、サポートされている 32 ビット Windows プラットフォーム上または Solaris 上の SQL Anywhere クライアント、およびサポートされている 32 ビット Windows プラットフォーム (Windows Mobile を含む) 上または UNIX 上の Ultra Light クライアントです。

- ***identity-file*** ID ファイルのパスとファイル名を指定します。この ID ファイルには、サーバーのプライベートキーとサーバーの証明書、さらにオプションで、認証局によって署名された証明書が格納されています。

  サーバー証明書の作成の詳細については、「デジタル証明書」1414 ページを参照してください。サーバー証明書は、自己署名証明書、または認証局やエンタープライズルート証明書の署名を受けた証明書のいずれかです。

- ***パスワード*** サーバーのプライベートキーのパスワードを指定します。このパスワードは、サーバー ID を作成するときに指定します。

### 参照

- 「-x mlsrv12 オプション」『Mobile Link サーバー管理』

### 例

次の例では、セキュリティのタイプ (RSA)、サーバー ID ファイル、サーバーのプライベートキーを保護する ID パスワードを mlsrv12 コマンドラインで指定します。

```
mlsrv12 -c "dsn=my_cons"
 -x tls(tls_type=rsa;identity=c:¥test¥serv_rsa1.crt;identity_password=pwd)
```

次の例では、mlsrv12 コマンドラインで ECC ID を指定します。

```
mlsrv12 -c "dsn=my_cons"
 -x tls(tls_type=ecc;identity=c:¥test¥serv_ecc1.crt;identity_password=pwd)
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

次の例は前の例と似ていますが、ID ファイル名にスペースが含まれる点だけが異なります。

```
mlsrv12 -c "dsn=my_cons"
 -x "tls(tls_type=rsa;identity=c:¥Program Files¥test¥serv_rsa1.crt;identity_password=pwd)"
```

サーバー ID ファイル (この例では *serv_ecc1.crt*) の作成については、「デジタル証明書」1414 ページを参照してください。

設定ファイルとファイル難読化ユーティリティ (dbfhide) を使用して、コマンドラインオプションを非表示にできます。「@data mlsrv12 オプション」『Mobile Link サーバー管理』を参照してください。

## トランスポートレイヤーセキュリティを使用する MobiLink クライアントの設定

Mobile Link トランスポートレイヤーセキュリティを使用するように SQL Anywhere クライアントまたは Ultra Light クライアントを設定できます。各クライアントに対して、信頼できる証明書、暗号化のタイプ、ネットワークプロトコルを指定します。

トランスポートレイヤーセキュリティを設定する手順の概要については、「トランスポートレイヤーセキュリティの設定」1412 ページを参照してください。

### サーバー認証

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

#### デジタル署名

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

●証明書で実行されるアルゴリズムが、ユニークな値またはハッシュを生成します。

●証明書への署名または認証局のプライベートキーを使用して、ハッシュが暗号化されます。

●デジタル署名と呼ばれる暗号化ハッシュが、証明書に埋め込まれます。

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

Mobile Link クライアントが Mobile Link サーバーにアクセスする場合、各クライアントがトランスポートレイヤーセキュリティを使用するように設定されていると、サーバーはその証明書のコピーをクライアントに送信します。クライアントは、証明書に含まれているサーバーのパブリックキーを使用して証明書のデジタル署名を復号化し、証明書の新しいハッシュを算出して、2 つの値を比較します。値が一致する場合は、サーバーの証明書の整合性が確認されます。

自己署名証明書の詳細については、「自己署名ルート証明書」1415 ページを参照してください。

エンタープライズルート証明書と認証局の詳細については、「証明書チェーン」1415 ページを参照してください。

### 証明書フィールドの確認

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

グローバル署名証明書の詳細については、「グローバル署名証明書」1417 ページを参照してください。

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

- **組織**　組織フィールドは、certificate_company Mobile Link クライアント接続パラメーターに対応します。「certificate_company」『Mobile Link クライアント管理』を参照してください。
- **組織単位**　組織単位フィールドは、certificate_unit Mobile Link クライアント接続パラメーターに対応します。「certificate_unit」『Mobile Link クライアント管理』を参照してください。
- **通称**　通称フィールドは、certificate_name Mobile Link クライアント接続パラメーターに対応します。「certificate_name」『Mobile Link クライアント管理』を参照してください。

Mobile Link クライアントの設定については、次の項を参照してください。


- 「トランスポートレイヤーセキュリティを使用する Ultra Light クライアントの設定」1433 ページ
- 「クライアントセキュリティオプション」1430 ページ


デジタル証明書の作成については、「デジタル証明書」1414 ページを参照してください。

## クライアントセキュリティオプション

Mobile Link クライアント (SQL Anywhere および Ultra Light) は、共通の接続パラメーターセットを使用して、トランスポートレイヤーセキュリティを設定します。

### trusted_certificate プロトコルオプション

Mobile Link クライアントは、trusted_certificate プロトコルオプションを使用して、信用された Mobile Link サーバー証明書を指定します。信頼できる証明書は、サーバーの自己署名証明書、パブリックエンタープライズルート証明書、民間認証局に属する証明書のいずれかです。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

次の項を参照してください。


- 「trusted_certificate」『Mobile Link クライアント管理』
- 「デジタル証明書」1414 ページ


### 証明書フィールドの確認

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

次の項を参照してください。


- 「証明書フィールドの確認」1430 ページ
- 「グローバル署名証明書」1417 ページ
- 「サーバー認証」1429 ページ


## トランスポートレイヤーセキュリティを使用する SQL Anywhere クライアントの設定

ここでは、HTTPS または TCP/IP でトランスポートレイヤーセキュリティを使用する SQL Anywhere クライアントの設定方法について説明します。

### TCP/IP および HTTPS におけるトランスポートレイヤーセキュリティの使用

Mobile Link トランスポートレイヤーセキュリティは、Mobile Link HTTPS および TCP/IP プロトコルの本来の機能です。HTTPS でトランスポートレイヤーセキュリティを使用するには、ADR 拡張オプションを使用して、trusted_certificate 接続パラメーターを指定します。dbmlsync コマンドラインの一部の構文を次に示します。

```
-e "ctp=protocol;
  adr=[ fips={ y | n }; ]
  trusted_certificate=public-certificate;
  ..."
```

- ***protocol*** 使用するプロトコルです。**https** または **tls** を指定します。**tls** で指定されるプロトコルはトランスポートレイヤーセキュリティを使用する TCP/IP です。

- **fips** FIPS 認定暗号化を使用するかどうかを示します。FIPS 認定暗号化は、RSA 暗号化のみと組み合わせて使用できます。FIPS 認定の HTTPS は、Certicom の FIPS 140-2 認定ソフトウェアという別のソフトウェアを使用しますが、HTTPS を使用するバージョン 9.0.2 以降の Mobile Link サーバーと互換性があります。

- ***public-certificate*** 信頼できる証明書のパスとファイル名を指定します。

  HTTPS または FIPS 認定の HTTPS の場合は、RSA 暗号化を使用して作成した証明書を使用してください。

### 参照

- 「クライアントセキュリティオプション」1430 ページ
- 「デジタル証明書」1414 ページ
- 「CommunicationAddress (adr) 拡張オプション」『Mobile Link クライアント管理』
- 「Mobile Link クライアントネットワークプロトコルオプション」『Mobile Link クライアント管理』
- 「CREATE SYNCHRONIZATION SUBSCRIPTION 文 [Mobile Link]」『SQL Anywhere サーバー SQL リファレンス』
- 「ALTER SYNCHRONIZATION SUBSCRIPTION 文 [Mobile Link]」『SQL Anywhere サーバー SQL リファレンス』

### 例

次の例では、HTTPS を使用して RSA セキュリティを指定します。コマンドは、全体を 1 行で入力する必要があります。

```
dbmlsync -c "server=rem1;uid=DBA;pwd=mypwd"
 -e "ctp=https;
     adr='trusted_certificate=c:¥temp¥public_cert.crt;
     certificate_company=Sybase, Inc.;
     certificate_unit=IAS;
     certificate_name=MobiLink'"
```

CREATE SYNCHRONIZATION SUBSCRIPTION 文または ALTER SYNCHRONIZATION SUBSCRIPTION 文を使用して、CommunicationAddress 拡張オプションを指定することもできます。この方法でも同じ情報が提供されますが、その情報はデータベースに保存されます。

```
CREATE SYNCHRONIZATION SUBSCRIPTION
 TO pub1
 FOR user1
 ADDRESS 'trusted_certificate=c:¥temp¥public_cert.crt;
   certificate_company=Sybase, Inc.;
   certificate_unit=IAS;
  certificate_name=MobiLink';
```

次の例では、RSA セキュリティと TCP/IP を指定します。コマンドは、全体を 1 行で入力する必要があります。

```
dbmlsync -c "server=rem1;uid=myuid;pwd=mypwd"
  -e "ctp=tls;
      adr='port=3333;
          tls_type=rsa;
          trusted_certificate=c:¥test¥public_cert.crt;
          certificate_company=Sybase, Inc.;
          certificate_unit=IAS;
          certificate_name=MobiLink'"
```

CREATE SYNCHRONIZATION SUBSCRIPTION 文または ALTER SYNCHRONIZATION SUBSCRIPTION 文を使用して、CommunicationAddress 拡張オプションを指定することもできます。

```
CREATE SYNCHRONIZATION SUBSCRIPTION
 TO pub1
 FOR user1
 ADDRESS 'port=3333;
   tls_type=rsa;trusted_certificate=public_cert.crt;
      certificate_company=Sybase, Inc.;
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
certificate_unit=IAS;
certificate_name=MobiLink';
```

## トランスポートレイヤーセキュリティを使用する Ultra Light クライアントの設定

**♦ TCP/IP または HTTPS でトランスポートレイヤーセキュリティを使用するように Ultra Light クライアントを設定するには、次の手順に従います。**

Mobile Link トランスポートレイヤーセキュリティは、Mobile Link HTTPS プロトコルの本来の機能です。HTTPS および Ultra Light クライアントを使用する場合は、信頼できる証明書と、ネットワークプロトコルオプションとして、証明書フィールドを直接指定できます。

Ultra Light インターフェイスで HTTPS プロトコルを指定する方法については、「Ultra Light 同期ストリームのネットワークプロトコルのオプション」『Ultra Light データベース管理とリファレンス』を参照してください。

tls_type 同期パラメーターの詳細については、「tls_type」『Mobile Link クライアント管理』を参照してください。

1. 信頼できるルート証明書を指定する方法は 2 つあります。
   - **Ultra Light データベースの作成時**　「Ultra Light データベース初期化ユーティリティ (ulinit)」『Ultra Light データベース管理とリファレンス』を参照してください。
   - **trusted_certificate プロトコルオプションの使用**　詳細については、手順 3 を参照してください。
2. TCP/IP または HTTPS プロトコルを同期用に指定します。セキュア TCP/IP のキーワードは tls です。

   次の例は C/C++ Ultra Light で記述されています。tls を指定するには、https を tls に変更します。

   ```
   auto ul_sync_info synch_info;
   conn.InitSynchInfo( &synch_info );
   synch_info.user_name = UL_TEXT( "50" );
   synch_info.version = UL_TEXT( "ul_default" );
   ...
   synch_info.stream = "https";
   ...
   ```
3. TCP/IP または HTTPS プロトコルオプションを指定します。

   次の例は C/C++ Ultra Light で記述されています。tls を指定するには、https を tls に変更します。

   ```
   auto ul_sync_info synch_info;
   ...
   synch_info.stream = "https";
   synch_info.stream_parms = TEXT(
        "port=9999;
         certificate_company=Sybase, Inc.;
         certificate_unit=IAS;
         certificate_name=MobiLink");
   ```

certificate_company、certificate_unit、certificate_name の各プロトコルオプションは、証明書のフィールドを確認するために使用されています。「証明書フィールドの確認」1430 ページを参照してください。

trusted_certificate HTTPS プロトコルオプションを指定することもできます。このオプションを指定すると、Ultra Light データベースに埋め込まれている、信用された証明書情報 (手順 1) が上書きされます。

```
auto ul_sync_info synch_info;
...
synch_info.stream = "https";
synch_info.stream_parms = TEXT(
    "port=9999;
    trusted_certificate=¥rsaroot.crt;
    certificate_company=Sybase, Inc.;
    certificate_unit=IAS;
    certificate_name=MobiLink");
```

HTTPS オプションの詳細については、「Ultra Light 同期ストリームのネットワークプロトコルのオプション」『Ultra Light データベース管理とリファレンス』を参照してください。

# 証明書ユーティリティ

ユーザーは、一般的にサードパーティから証明書を購入します。そのような認証局には、証明書を作成するために独自のツールが用意されています。次のツールは、開発用やテスト用の証明書を作成するときに特に便利ですが、稼働時の証明書に使用することもできます。

**参照**


- 「証明書作成ユーティリティ (createcert)」924 ページ
- 「証明書ビューアーユーティリティ (viewcert)」934 ページ


 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

# レプリケーション

この項では、SQL Anywhere を Open Server として使用する方法について説明します。

SQL Anywhere でサポートされている同期／レプリケーションテクノロジの詳細については、「同期テクノロジの比較」『SQL Anywhere 12 紹介』を参照してください。

## Open Server としての SQL Anywhere

SQL Anywhere は、クライアントアプリケーションにとっては Open Server として機能します。この機能を使用すると、Sybase Open Client アプリケーションは SQL Anywhere データベースにネイティブに接続できます。

単に、Sybase のアプリケーションを SQL Anywhere とともに使用するだけの場合は、Sybase Open Client、Sybase Open Server、または TDS の詳細を知る必要はありません。しかし、これらのコンポーネントがどのように互いに適合するかを理解すれば、データベースの設定やアプリケーションの設定に役立ちます。この項では、これらのコンポーネントが互いにどう適合するかを説明しますが、それぞれの内部機能についての説明は省略します。

### Sybase Open Client と Sybase Open Server

SQL Anywhere とその他の Adaptive Server ファミリのメンバーは、「Open Server」として動作します。つまり、Sybase から入手可能な「Sybase Open Client」 ライブラリを使用して、クライアントアプリケーションを開発できます。Sybase Open Client には、Client Library (CT-Library) インターフェイス、旧式の DB-Library インターフェイスの両方が含まれています。

### Tabular Data Stream

Sybase Open Client と Sybase Open Server は、「Tabular Data Stream」 (TDS) と呼ばれるアプリケーションプロトコルを使用して情報を交換します。Sybase Open Client ライブラリを使用して構築されたすべてのアプリケーションは、TDS アプリケーションでもあります。これは、Sybase Open Client ライブラリが TDS インターフェイスを使用するためです。ただし、(jConnect などの) 一部のアプリケーションは、Sybase Open Client ライブラリを使用しませんが、TDS アプリケーションです。これらのアプリケーションは、TDS プロトコルを使用して直接通信します。

多くの Open Server が Sybase Open Server ライブラリを使用して TDS へのインターフェイスを処理していますが、固有の TDS への直接インターフェイスを持つアプリケーションもあります。Sybase の Adaptive Server Enterprise と SQL Anywhere には、両方とも内部 TDS インターフェイスがあります。両方ともクライアントアプリケーションには Open Server として表示されますが、Sybase Open Server ライブラリを使用しません。

### プログラミングインターフェイスとアプリケーションプロトコル

SQL Anywhere は 2 種類のアプリケーションプロトコルをサポートします。Sybase Open Client アプリケーションとその他の Sybase アプリケーション (OmniConnect など) では、TDS を使用します。ODBC アプリケーションと Embedded SQL アプリケーションでは、それぞれ個別に、SQL Anywhere に特有のアプリケーションプロトコルを使用します。

### TDS による TCP/IP の使用

TDS のようなアプリケーションプロトコルは、ネットワークトラフィックを処理する下位レベルの通信プロトコルの一番上に位置します。SQL Anywhere は、TCP/IP ネットワークプロトコル上でのみ TDS をサポートします。それに対し、SQL Anywhere 固有のアプリケーションプロトコルは、さまざまなネットワークプロトコルと、同一コンピューターでの通信用に設計された共有メモリプロトコルをサポートします。

## Sybase アプリケーションと SQL Anywhere

SQL Anywhere は Open Server として動作できるので、OmniConnect などの Sybase アプリケーションを SQL Anywhere とともに動作させることができます。

### OmniConnect のサポート

Sybase OmniConnect は、データの内容やデータの所在がわからなくても複数のデータソースにアクセスすることにより、企業内でデータを統合して表示することができます。さらに、OmniConnect は、企業全体にわたってデータの異機種間ジョインを実行し、IBM DB2、Sybase Adaptive Server Enterprise、Oracle、VSAM のようなターゲットについて、プラットフォームを問わないテーブルジョインを可能にします。

Open Server インターフェイスを使用すると、SQL Anywhere を OmniConnect 用のデータソースとして使用できます。

## Open Server としての SQL Anywhere の設定

この項では、Sybase Open Client アプリケーションからの接続を受信するように SQL Anywhere サーバーを設定する方法を説明します。

## システムの稼働条件

SQL Anywhere を Open Server として使用するには、クライアント側とサーバー側でそれぞれ別の稼働条件があります。

### サーバー側の稼働条件

SQL Anywhere を Open Server として使用するには、サーバー側に次の要素が必要です。

- **SQL Anywhere サーバーコンポーネント**　ネットワークを介して Open Server にアクセスする場合は、ネットワークサーバー (*dbsrv12.exe*) を使用します。パーソナルサーバー (*dbeng12.exe*) を Open Server として使用できるのは、同一コンピューターからの接続に限られます。

- **TCP/IP**　ネットワーク接続をしていない場合でも、SQL Anywhere を Open Server として使用するには、TCP/IP プロトコルスタックが必要です。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

### クライアント側の稼働条件

Sybase クライアントアプリケーションを使用して SQL Anywhere を含む Open Server に接続するには、次の要素が必要です。

- **Sybase Open Client コンポーネント**　アプリケーションが Sybase Open Client を使用する場合、TDS 経由でアプリケーションが通信するために必要なネットワークライブラリは、Sybase Open Client ライブラリによって提供されます。

- **jConnect**　アプリケーションが JDBC を使用する場合は、jConnect と Java ランタイム環境が必要です。jConnect をダウンロードするには、http://www.sybase.com/products/allproductsa-z/softwaredeveloperkit/jconnect に移動してください。

- **DSEdit**　サーバー名を Sybase Open Client アプリケーションから使用できるようにするには、ディレクトリサービスエディター DSEdit が必要です。UNIX プラットフォームでは、このユーティリティは sybinit と呼ばれます。

  DSEdit は、SQL Anywhere には付属していませんが、Sybase Open Server ソフトウェアには付属しています。

## Open Server としてのデータベースサーバーの設定

SQL Anywhere を Open Server として使用する場合は、TCP/IP プロトコルを使用して起動してください。デフォルトでは、使用可能なすべての通信プロトコルがサーバーによって起動されますが、起動されるプロトコルをコマンドに明示的にリストすることによって、それらのプロトコルを制限できます。たとえば、次のコマンドは TCP/IP 通信プロトコルを開始します。

```
dbsrv12 -x tcpip -n myserver c:¥mydata.db
```

パーソナルデータベースサーバーは TCP/IP プロトコルをサポートするため、このサーバーを、同一コンピューター上の通信用 Open Server として使用できます。

このサーバーは、TDS 経由で Sybase Open Client アプリケーションにサービスすると同時に、SQL Anywhere 専用のアプリケーションプロトコルを使用して、TCP/IP プロトコルなどのプロトコル経由で他のアプリケーションにサービスを提供します。

### Sybase Open Client の設定値

このサーバーに接続するには、データベースサーバーを実行しているコンピューター名とそのサーバーが使用している TCP/IP ポートをクライアントコンピューター側の interfaces ファイルに指定します。

クライアントコンピューターの設定の詳細については、「Sybase Open Server の設定」1438 ページを参照してください。

### 参照

- ServerPort 接続プロパティ 743 ページ

# Sybase Open Server の設定

SQL Anywhere は、ネットワーク上の他の Adaptive Server アプリケーション、Open Server アプリケーション、クライアントソフトウェアと通信できます。クライアントは 1 つ以上のサーバーと通信でき、サーバーはリモートプロシージャーコールを通して他のサーバーと通信できます。製品が互いに対話するには、他の製品がネットワークのどこに常駐しているかをそれぞれが認識する必要があります。このネットワークサービス情報は interfaces ファイルに格納されています。

## interfaces ファイル

「interfaces ファイル」の名前は通常、Windows オペレーティングシステムでは *SQL.ini*、UNIX オペレーティングシステムでは *interfaces* または *interfac* です。

interfaces ファイルは、住所録のようなもので、コンピューター上の Sybase Open Client アプリケーションが認識しているあらゆるデータベースサーバーの名前とアドレスが列記されています。Sybase Open Client プログラムを使用してデータベースサーバーに接続すると、プログラムは interfaces ファイルでサーバーの名前を検索し、そのアドレスを使用してサーバーに接続します。

interfaces ファイルの名前、場所、内容はオペレーティングシステムによって異なります。また、interfaces ファイル中のアドレスのフォーマットもネットワークプロトコルによって異なります。

SQL Anywhere をインストールすると、インストーラーによって簡単な interfaces ファイルが作成されます。このファイルを使用して、TCP/IP 経由で SQL Anywhere にローカル接続できます。システム管理者の役割は、interfaces ファイルを修正してユーザーに配布し、ユーザーがネットワークを通じて SQL Anywhere に接続できるようにすることです。

## DSEdit ユーティリティ

DSEdit ユーティリティは Windows ユーティリティです。このユーティリティを使用して interfaces ファイル (*SQL.ini*) を設定できます。以下の項では、DSEdit ユーティリティを使用して interfaces ファイルを設定する方法を説明します。

以下の項では、SQL Anywhere に必要なタスクのために DSEdit を使用する方法を説明します。これは、DSEdit ユーティリティの完全なマニュアルではありません。

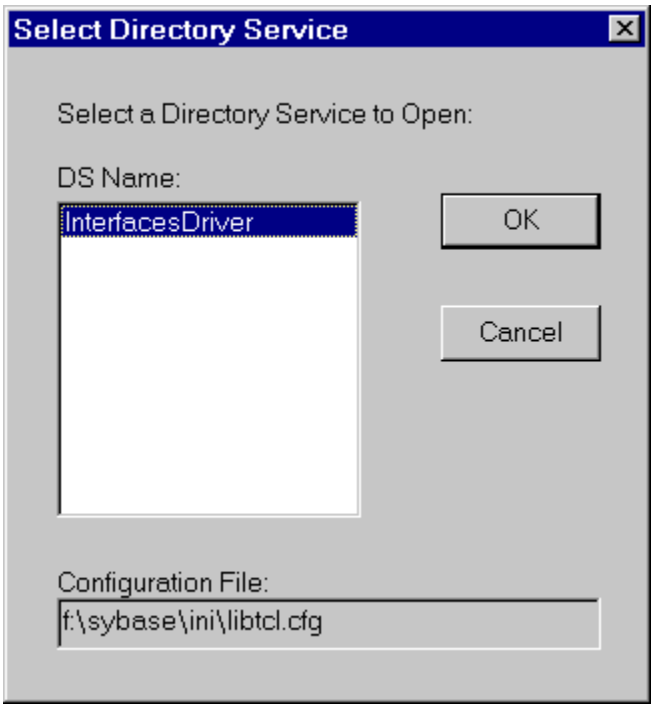
DSEdit の詳細については、他の Sybase 製品に同梱されているプラットフォーム別の『**設定ガイド**』を参照してください。

## DSEdit の起動

DSEdit 実行プログラムは、*SYBASE¥bin* フォルダーにあります。このディレクトリは、インストール時にパスに追加されます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

DSEdit を起動すると、**[Select Directory Service]** ウィンドウが表示されます。




## ディレクトリサービスのセッションを開く

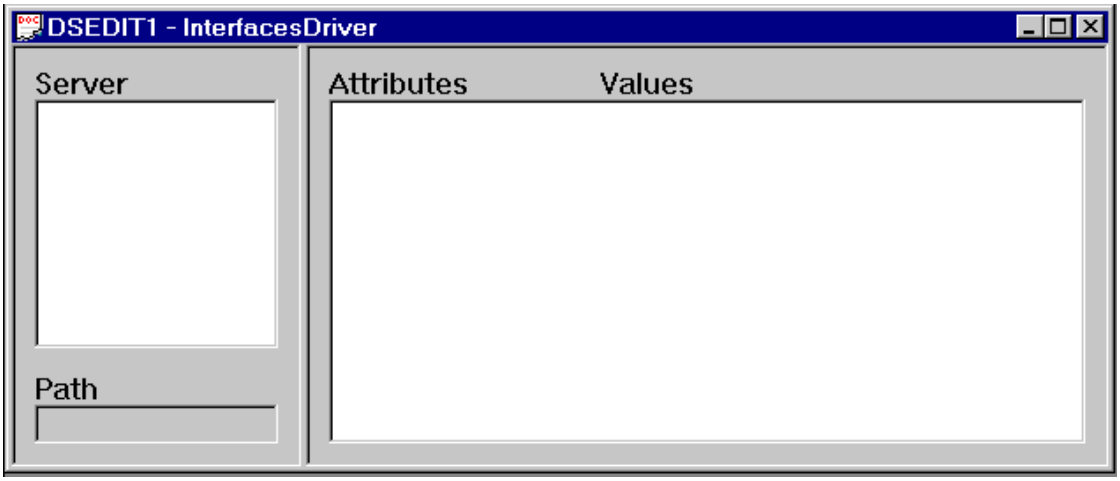
**♦ ディレクトリサービスのセッションを開くには、次の手順に従います。**

**[Select Directory Service]** ウィンドウで、ディレクトリサービスのセッションを開きます。セッションを開くと、interfaces ファイル (*SQL.ini*)、または *libtcl.cfg* ファイルにリストされたドライバーを持つディレクトリサービスを編集できます。

> **注意**
> SYBASE 環境変数を設定する必要があります。DSEdit ユーティリティは、SYBASE 環境変数を使用して *libtcl.cfg* ファイルを検索します。SYBASE 環境変数が正しくない場合、DSEdit は *libtcl.cfg* ファイルを見つけることができません。

- **[DS Name]** リストから、接続先とするディレクトリサービスのローカル名をクリックし、**[OK]** をクリックします。

  SQL Anywhere の場合は、**[InterfacesDriver]** をクリックしてください。**[InterfacesDriver]** ウィンドウで、SQL Anywhere サーバーなどのサーバーについて、エントリの追加、修正、削除ができます。




## サーバーエントリの追加

♦ **サーバーエントリを追加するには、次の手順に従います。**

1. **[サーバーオブジェクト]** メニューで、**[追加]** をクリックします。
2. **[Server Name]** ボックスにサーバー名を入力し、**[OK]** をクリックします。

   入力するサーバー名は、接続先のデータベース名と一致する必要があります。サーバーアドレスは、サーバーの名前と場所を指定するために使用されます。**[Server Name]** フィールドは、Sybase Open Client の識別子です。SQL Anywhere では、データベースサーバーに複数のデータベースがロードされている場合、DSEdit のサーバー名エントリによって使用するデータベースが識別されます。

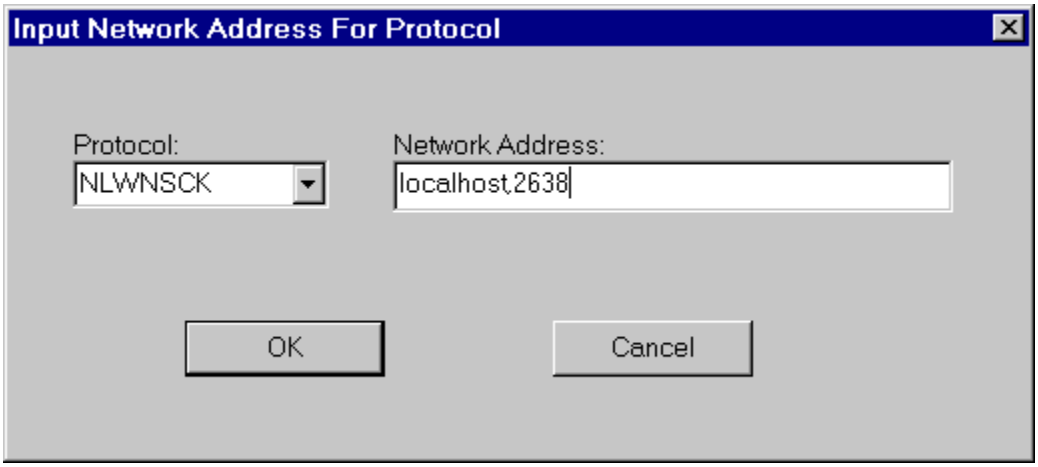
   サーバーエントリが **[Server]** ボックスに表示されます。サーバーの属性を指定するには、エントリを修正します。

## サーバーアドレスの追加または変更

♦ **サーバーアドレスを入力するには、次の手順に従います。**

**[Server Name]** を入力したら、**[Server Address]** を変更して interfaces ファイルの入力を完了します。

1. **[Server]** ボックスで、サーバーエントリをクリックします。
2. **[Attributes]** ボックスでサーバーアドレスを右クリックし、**[Modify Attribute]** をクリックします。
3. **[Add]** をクリックします。
4. **[Protocol]** リストで、TCP/IP プロトコルの **[NLWNSCK]** をクリックします。




5. **[Network Address]** フィールドに、有効なネットワークアドレスを入力します。TCP/IP アドレスは、次のいずれかの形式で指定します。

   ●コンピューター名、ポート番号

   ●IP アドレス、ポート番号

   アドレス (またはコンピューター名) とポート番号をカンマで区切ります。

   **コンピューター名**　サーバーを実行しているコンピューターは、名前 (または IP アドレス) によって識別されます。Windows オペレーティングシステムの場合、[コントロール パネル] の [ネットワーク] でコンピューター名を検索できます。

   クライアントとサーバーが同一のコンピューター上にある場合でも、コンピューター名を入力します。この場合は、**localhost** を使用して現在のコンピューターを識別できます。

   **ポート番号**　入力するポート番号は、SQL Anywhere データベースサーバーを起動するために使用したポート番号と一致させます。「Open Server としてのデータベースサーバーの設定」1437 ページを参照してください。

   SQL Anywhere サーバーのデフォルトのポート番号は 2638 です。この番号は、Internet Adapter Number Authority によって SQL Anywhere に割り当てられており、明示的に他のポートを使用する正当な理由がないかぎり、このポートを使用することをおすすめします。

   次に示すのは、有効なサーバーアドレスエントリです。

   ```
   elora,2638
   123.85.234.029,2638
   ```

6. **[OK]** をクリックします。

## サーバーアドレスの確認

♦ **サーバーを ping してサーバーアドレスを検証するには、次の手順に従います。**

> **注意**
> データベース接続は検証されません。ネットワーク接続を検証すると、指定されたコンピューター名とポート番号でサーバーが要求を受信していることが確認されます。データベース接続については何も確認されません。

1. データベースサーバーが実行していることを確認します。
2. **DSEdit** セッションウィンドウの **[Server]** ボックスで、サーバーエントリをクリックします。
3. **[Server Object]** » **[Ping Server]** をクリックします。
4. ping するアドレスを選択し、**[Ping]** をクリックします。

   ウィンドウが表示され、接続できたかどうかが通知されます。接続できたことが表示された場合、オープン接続とクローズ接続の両方に成功したことを意味します。

## サーバーエントリ名の変更

♦ **サーバーエントリ名を変更するには、次の手順に従います。**

**DSEdit** セッションウィンドウからサーバーエントリ名を変更できます。

1. **[Server]** ボックスで、サーバーエントリをクリックします。
2. **[Server Object]** » **[Rename]** をクリックします。
3. **[Server Name]** ボックスに新しいサーバーエントリ名を入力します。
4. **[OK]** をクリックします。

## サーバーエントリの削除

♦ **サーバーエントリを削除するには、次の手順に従います。**

**DSEdit** セッションウィンドウからサーバーエントリを削除できます。

1. **[Server]** ボックスで、サーバーエントリをクリックします。
2. **[Server Object]** » **[削除]** をクリックします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

## JDBC のサーバーの設定

JDBC 接続のアドレス (URL) には、サーバー検索用の必要な情報がすべて含まれています。「ドライバーへの URL の指定」『SQL Anywhere サーバー プログラミング』を参照してください。

# Sybase Open Client と jConnect 接続の特性

SQL Anywhere は、TDS を通してアプリケーションをサービスしている場合、関連したデータベースのさまざまなオプションを自動的に設定し、オプションの値を Adaptive Server Enterprise のデフォルトの設定と互換性があるようにします。このオプションの設定は、その接続中だけの一時的なものです。オプションは、クライアントアプリケーションによっていつでも無効にできます。

### デフォルトの設定値

TDS を使用する接続で設定されるデータベースオプションは、次のとおりです。

| オプション | 設定値 |
|---|---|
| allow_nulls_by_default | Off |
| ansi_blanks | On |
| ansinull | Off |
| chained | Off |
| close_on_endtrans | Off |
| date_format | YYYY-MM-DD |
| date_order | MDY |
| escape_character | Off |
| isolation_level | 1 |
| on_tsql_error | Continue |
| quoted_identifier | Off |
| time_format | HH:NN:SS.SSS |
| timestamp_format | YYYY-MM-DD HH:NN:SS.SSS |
| tsql_variables | On |

### 起動オプションの設定方法

TDS 接続用のデフォルトデータベースオプションは、システムプロシージャー sp_tsql_environment を使用して設定されます。このプロシージャーでは、以下のオプションを設定します。

```
SET TEMPORARY OPTION allow_nulls_by_default='Off';
SET TEMPORARY OPTION ansi_blanks='On';
SET TEMPORARY OPTION ansinull='Off';
SET TEMPORARY OPTION chained='Off';
SET TEMPORARY OPTION close_on_endtrans='Off';
SET TEMPORARY OPTION date_format='YYYY-MM-DD';
SET TEMPORARY OPTION date_order='MDY';
SET TEMPORARY OPTION escape_character='Off';
SET TEMPORARY OPTION isolation_level='1';
SET TEMPORARY OPTION on_tsql_error='Continue';
SET TEMPORARY OPTION quoted_identifier='Off';
SET TEMPORARY OPTION time_format='HH:NN:SS.SSS';
SET TEMPORARY OPTION timestamp_format='YYYY-MM-DD HH:NN:SS.SSS';
SET TEMPORARY OPTION tsql_variables='On';
```

**注意**
sp_tsql_environment プロシージャーは、変更しないでください。このプロシージャーは、システムが専用に使用します。

このプロシージャーは、TDS 通信プロトコルを使用する接続についてのみ、オプションを設定します。設定される接続には、jConnect を使用する Sybase Open Client 接続や JDBC 接続があります。その他の接続 (ODBC と Embedded SQL) は、データベース用のデフォルト設定値を持っています。

TDS 接続用のオプションは、次のようにして変更できます。

**♦ TDS 接続用オプションの設定値を変更するには、次の手順に従います。**

1. 目的のデータベースオプションを設定するプロシージャーを作成します。たとえば、次のようなプロシージャーを使用できます。

   ```
   CREATE PROCEDURE my_startup_procedure()
   BEGIN
     IF CONNECTION_PROPERTY('CommProtocol')='TDS' THEN
       SET TEMPORARY OPTION quoted_identifier='Off';
     END IF
   END;
   ```

   このプロシージャーの例では、デフォルト設定の quoted_identifier オプションだけを変更します。

2. login_procedure オプションに新しいプロシージャー名を設定します。

   ```
   SET OPTION login_procedure= 'DBA.my_startup_procedure';
   ```

   今後の接続では、このプロシージャーを使用します。別のユーザー ID に対して、別のプロシージャーを設定できます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

**参照**

- 「データベースオプション」575 ページ

# 索引

## 記号


-? サーバーオプション
  Windows Mobile でサポート対象外, 438
  データベースサーバーオプション (dbeng12、dbsrv12), 203
.dbr
  説明, 1060
.logr
  説明, 1060
.NET Compact Framework
  SQL Anywhere for Windows Mobile で使用, 405
.odbc.ini
  DSN 接続パラメーターでの指定, 338
  Mac OS X でのデータソースの作成, 45
  暗号化されたパスワードの保管, 341
  説明, 48
  データソースの作成, 937
  パスワードの保管, 360
.pid
  Linux サービス, 1011
@data オプション
  Broadcast Repeater ユーティリティ (dbns12), 921
  Interactive SQL ユーティリティ (dbisql), 976
  ping ユーティリティ(dbping), 992
  SQL Anywhere コンソールユーティリティ (dbconsole), 1021
  Windows Mobile でサポート対象外, 438
  Windows サービスユーティリティ (dbsvc), 1013
  アップグレードユーティリティ (dbupgrad), 1062
  アンロードユーティリティ (dbunload), 1044
  検証ユーティリティ (dbvalid), 1064
  サポートユーティリティ (dbsupport), 1027
  サーバーライセンス取得ユーティリティ (dblic), 1003
  サーバー列挙ユーティリティ (dblocate), 999
  消去ユーティリティ (dberase), 951
  情報ユーティリティ (dbinfo), 957
  初期化ユーティリティ (dbinit), 959
  説明, 909
  停止ユーティリティ (dbstop), 1024
  データソースユーティリティ (dbdsn), 936
  データベースサーバーオプション (dbeng12、dbsrv12), 201
  トランザクションログユーティリティ (dblog), 1041
  バックアップユーティリティ (dbbackup), 913
  ヒストグラムユーティリティ (dbhist), 955
  プロセス生成ユーティリティ (dbspawn), 1023
  ログ変換ユーティリティ (dbtran), 985
&
  UNIX コマンドライン, 180
  設定ファイルでの使用, 909
#
  設定ファイルでの使用, 909
0 埋め込み
  date_format オプションによる制御, 614
  sr_time_format オプションによる制御, 682
  sr_timestamp_format オプションによる制御, 684
  sr_timestamp_with_time_zone_format オプションによる制御, 686
  time_format オプションによる制御, 697
  timestamp_format オプションによる制御, 699
  timestamp_with_time_zone_format オプションによる制御, 701
10 進数精度
  データベースオプション, 661
1254TRKALT 照合
  1254TRKALT の違い, 511
  使用, 511
32 ビット
  PATH 環境変数, 447
4 ギガバイトのチューニング
  データベースサーバーキャッシュ, 219
7 ビット文字
  説明, 474
-ac オプション
  アンロードユーティリティ (dbunload), 1044
  サポートユーティリティ (dbsupport), 1036
-ad オプション
  データベースオプション (dbeng12、dbsrv12), 306
-af オプション
  サポートユーティリティ (dbsupport), 1036
  初期化ユーティリティ (dbinit), 959
-an オプション
  アンロードユーティリティ (dbunload), 1044
-ap オプション

Broadcast Repeater ユーティリティ (dbns12), 921
アンロードユーティリティ (dbunload), 1044
-ar オプション
アンロードユーティリティ (dbunload), 1044
データベースオプション (dbeng12、dbsrv12), 307
-as オプション
Linux サービスユーティリティ (dbsvc), 1008
Windows サービスユーティリティ (dbsvc), 1016
データベースオプション (dbeng12、dbsrv12), 308
-a オプション
Linux サービスユーティリティ (dbsvc), 1008
Windows サービスユーティリティ (dbsvc), 1016
初期化ユーティリティ (dbinit), 959
データベースオプション (dbeng12、dbsrv12), 305
ログ変換ユーティリティ (dbtran), 985
-b オプション
証明書作成ユーティリティ (createcert), 924
初期化ユーティリティ (dbinit), 959
データソースユーティリティ (dbdsn), 937
データベースサーバーオプション (dbeng12、dbsrv12), 203
バックアップユーティリティ (dbbackup), 913
-ca オプション
証明書作成ユーティリティ (createcert), 924
データベースサーバーオプション (dbeng12、dbsrv12), 206
-cc オプション
サポートユーティリティ (dbsupport), 1032
データベースサーバーオプション (dbeng12、dbsrv12), 207
-cd オプション
サポートユーティリティ (dbsupport), 1032
-cet オプション
サポートユーティリティ (dbsupport), 1032
-ce オプション
サポートユーティリティ (dbsupport), 1032
-chx オプション
データベースサーバーオプション (dbeng12、dbsrv12), 209
-ch オプション
サポートユーティリティ (dbsupport), 1032
データベースサーバーオプション (dbeng12、dbsrv12), 208
-cid オプション
サポートユーティリティ (dbsupport), 1032
-ck オプション
証明書作成ユーティリティ (createcert), 924
-cl オプション
データソースユーティリティ (dbdsn), 936
データベースサーバーオプション (dbeng12、dbsrv12), 211
-cm オプション
Linux サービスユーティリティ (dbsvc), 1011
Windows Mobile でサポート対象外, 438
Windows サービスユーティリティ (dbsvc), 1019
アンロードユーティリティ (dbunload), 1044
データソースユーティリティ (dbdsn), 937
データベースサーバーオプション (dbeng12、dbsrv12), 212
-co オプション
証明書作成ユーティリティ (createcert), 924
-cp オプション
アンロードユーティリティ (dbunload), 1044
サポートユーティリティ (dbsupport), 1032
証明書作成ユーティリティ (createcert), 924
データベースサーバーオプション (dbeng12、dbsrv12), 213
-cr オプション
サポートユーティリティ (dbsupport), 1032
データベースサーバーオプション (dbeng12、dbsrv12), 214
-cs オプション
データベースサーバーオプション (dbeng12、dbsrv12), 215
-cv オプション
データベースサーバーオプション (dbeng12、dbsrv12), 216
-cw オプション
Windows Mobile でサポート対象外, 438
データソースユーティリティ (dbdsn), 939
データベースサーバーオプション (dbeng12、dbsrv12), 217
-c オプション
dbisqlc ユーティリティ, 950
Interactive SQL ユーティリティ (dbisql), 976
ping ユーティリティ (dbping), 994
SQL Anywhere コンソールユーティリティ (dbconsole), 1021

SQL Anywhere スクリプト実行ユーティリティ (dbrunsql), 997
アップグレードユーティリティ (dbupgrad), 1062
アンロードユーティリティ (dbunload), 1044
検証ユーティリティ (dbvalid), 1064
情報ユーティリティ (dbinfo), 957
証明書作成ユーティリティ (createcert), 924
初期化ユーティリティ (dbinit), 959
接続文字列, 31
停止ユーティリティ (dbstop), 1024
データソースユーティリティ (dbdsn), 939
データベースサーバーオプション (dbeng12、dbsrv12), 204
バックアップユーティリティ (dbbackup), 913
ヒストグラムユーティリティ (dbhist), 955
ログ変換ユーティリティ (dbtran), 985
-d1 オプション
Interactive SQL ユーティリティ (dbisql), 976
-datasource オプション
Interactive SQL ユーティリティ (dbisql), 976
SQL Anywhere コンソールユーティリティ (dbconsole), 1021
-dba オプション
初期化ユーティリティ (dbinit), 959
-dbs オプション
初期化ユーティリティ (dbinit), 959
-dc オプション
アンロードユーティリティ (dbunload), 1044
-dh オプション
データベースオプション (dbeng12、dbsrv12), 309
-dn オプション
サーバー列挙ユーティリティ (dblocate), 999
-dr オプション
データソースユーティリティ (dbdsn), 937
-ds オプション
データベースオプション (dbeng12、dbsrv12), 308
-dt オプション
データベースサーバーオプション (dbeng12、dbsrv12), 221
-dv オプション
サーバー列挙ユーティリティ (dblocate), 999
-d オプション
dbisqlc ユーティリティ, 950
Interactive SQL ユーティリティ (dbisql), 976
Linux サービスユーティリティ (dbsvc), 1007
ping ユーティリティ (dbping), 994
SQL Anywhere スクリプト実行ユーティリティ (dbrunsql), 997
Windows サービスユーティリティ (dbsvc), 1013
アンロードユーティリティ (dbunload), 1044
検証ユーティリティ (dbvalid), 1064
サーバー列挙ユーティリティ (dblocate), 999
証明書ビューアーユーティリティ (viewcert), 934
停止ユーティリティ (dbstop), 1024
データソースユーティリティ (dbdsn), 936
バックアップユーティリティ (dbbackup), 913
ログ変換ユーティリティ (dbtran), 985
-ea オプション
アンロードユーティリティ (dbunload), 1044
初期化ユーティリティ (dbinit), 959
-ec オプション
クライアント／サーバー通信の保護, 1420
証明書作成ユーティリティ (createcert), 924
データベースサーバーオプション (dbeng12、dbsrv12), 223
-ek オプション
アンロードユーティリティ (dbunload), 1044
消去ユーティリティ (dberase), 951
初期化ユーティリティ (dbinit), 959
データベースオプション (dbeng12、dbsrv12), 310
トランザクションログユーティリティ (dblog), 1041
ログ変換ユーティリティ (dbtran), 985
-en オプション
ping ユーティリティ (dbping), 994
-ep オプション
アンロードユーティリティ (dbunload), 1044
消去ユーティリティ (dberase), 951
初期化ユーティリティ (dbinit), 959
データベースサーバーオプション (dbeng12、dbsrv12), 226
トランザクションログユーティリティ (dblog), 1041
ログ変換ユーティリティ (dbtran), 985
-er オプション
アンロードユーティリティ (dbunload), 1044
-es オプション
データベースサーバーオプション (dbeng12、dbsrv12), 227
-et オプション

アンロードユーティリティ (dbunload), 1044
初期化ユーティリティ (dbinit), 959
-e オプション
SQL Anywhere スクリプト実行ユーティリティ (dbrunsql), 997
アンロードユーティリティ (dbunload), 1044
サポートユーティリティ (dbsupport), 1028
-fc オプション
データベースサーバーオプション (dbeng12、dbsrv12), 229
-fips オプション
AES_FIPS 暗号化アルゴリズム, 230
AES256_FIPS 暗号化アルゴリズム, 230
データベースサーバーオプション (dbeng12、dbsrv12), 230
-fx オプション
検証ユーティリティ (dbvalid), 1064
-f オプション
Interactive SQL ユーティリティ (dbisql), 976
SQL Anywhere スクリプト実行ユーティリティ (dbrunsql), 997
データソースユーティリティ (dbdsn), 937
データベースサーバーオプション (dbeng12、dbsrv12), 228
プロセス生成ユーティリティ (dbspawn), 1023
ログ変換ユーティリティ (dbtran), 985
-ga オプション
データベースサーバーオプション (dbeng12、dbsrv12), 231
-gb オプション
Windows Mobile でサポート対象外, 438
データベースサーバーオプション (dbeng12、dbsrv12), 232
-gc オプション
データベースサーバーオプション (dbeng12、dbsrv12), 232
-gd オプション
データベースサーバーオプション (dbeng12、dbsrv12), 233
-ge オプション
Windows Mobile でサポート対象外, 438
データベースサーバーオプション (dbeng12、dbsrv12), 234
-gf オプション
データベースサーバーオプション (dbeng12、dbsrv12), 235
-gk オプション
データベースサーバーオプション (dbeng12、dbsrv12), 235
-gl オプション
データベースサーバーオプション (dbeng12、dbsrv12), 236
-gm オプション
データベースサーバーオプション (dbsrv12), 237
-gna オプション
Windows Mobile でサポート対象外, 438
クエリ内並列処理への影響, 238
データベースサーバーオプション (dbsrv12), 238
-gnh オプション
Windows Mobile でサポート対象外, 438
クエリ内並列処理への影響, 239
データベースサーバーオプション (dbsrv12), 239
-gnl オプション
Windows Mobile でサポート対象外, 438
クエリ内並列処理への影響, 240
データベースサーバーオプション (dbsrv12), 240
-gns オプション
Windows Mobile でサポート対象外, 438
データベースサーバーオプション (dbsrv12), 241
-gn オプション
データベースサーバーオプション (dbsrv12), 237
データベースサーバーの使用法, 164
データベースサーバーのマルチプログラミングレベル, 165
-gp オプション
データベースサーバーオプション (dbeng12、dbsrv12), 242
-gr オプション
データベースサーバーオプション (dbeng12、dbsrv12), 242
-gss オプション
データベースサーバーオプション (dbeng12、dbsrv12), 243
データベースサーバーの使用法, 164
-gtc オプション
Windows Mobile でサポート対象外, 438
データベースサーバーオプション (dbeng12、dbsrv12), 245
データベースサーバーの使用法, 164

-gt オプション
Windows Mobile でサポート対象外, 438
データベースサーバーオプション (dbeng12、dbsrv12), 244
データベースサーバーの使用法, 164
-gu オプション
データベースサーバーオプション (dbeng12、dbsrv12), 247
ユーティリティデータベースの文実行パーミッションの制御, 561
-g オプション
Linux サービスユーティリティ (dbsvc), 1007
SQL Anywhere スクリプト実行ユーティリティ (dbrunsql), 997
Windows サービスユーティリティ (dbsvc), 1013
アンロードユーティリティ (dbunload), 1044
データソースユーティリティ (dbdsn), 936
ログ変換ユーティリティ (dbtran), 985
-host オプション
Interactive SQL ユーティリティ (dbisql), 976
SQL Anywhere コンソールユーティリティ (dbconsole), 1021
-ii オプション
アンロードユーティリティ (dbunload), 1044
-im オプション
インメモリモード, 160
データベースサーバーオプション (dbeng12、dbsrv12), 247
-io オプション
証明書作成ユーティリティ (createcert), 924
-ip オプション
証明書ビューアーユーティリティ (viewcert), 934
-ir オプション
トランザクションログユーティリティ (dblog), 1041
ログ変換ユーティリティ (dbtran), 985
-is オプション
サポートユーティリティ (dbsupport), 1028
トランザクションログユーティリティ (dblog), 1041
ログ変換ユーティリティ (dbtran), 985
-it オプション
ログ変換ユーティリティ (dbtran), 985
-iu オプション
サポートユーティリティ (dbsupport), 1028
-ix オプション
アンロードユーティリティ (dbunload), 1044
-i オプション
Windows サービスユーティリティ (dbsvc), 1016
アップグレードユーティリティ (dbupgrad), 1062
検証ユーティリティ (dbvalid), 1064
初期化ユーティリティ (dbinit), 959
-j オプション
ログ変換ユーティリティ (dbtran), 985
-kd オプション
アンロードユーティリティ (dbunload), 1044
-kl オプション
データベースサーバーオプション (dbeng12、dbsrv12), 250
-ko オプション
証明書作成ユーティリティ (createcert), 924
-kp オプション
証明書作成ユーティリティ (createcert), 924
データベースサーバーオプション (dbeng12、dbsrv12), 251
-krb オプション
データベースサーバーオプション (dbeng12、dbsrv12), 253
-kr オプション
データベースサーバーオプション (dbeng12、dbsrv12), 252
-ksc オプション
データベースサーバーオプション (dbeng12、dbsrv12), 255
-ksd オプション
データベースサーバーオプション (dbeng12、dbsrv12), 255
-ks オプション
データベースサーバーオプション (dbeng12、dbsrv12), 254
-k オプション
アンロードユーティリティ (dbunload), 1044
サーバーライセンス取得ユーティリティ (dblic), 1003
初期化ユーティリティ (dbinit), 959
データベースサーバーオプション (dbeng12、dbsrv12), 249
バックアップユーティリティ (dbbackup), 913
ログ変換ユーティリティ (dbtran), 985
-lc オプション
サポートユーティリティ (dbsupport), 1028
-le オプション

初期化ユーティリティ (dbinit), 959
-ls オプション
サポートユーティリティ (dbsupport), 1028
-l オプション
Linux サービスユーティリティ (dbsvc), 1007
ping ユーティリティ (dbping), 994
Windows サービスユーティリティ (dbsvc), 1013
アンロードユーティリティ (dbunload), 1044
サーバーライセンス取得ユーティリティ (dblic), 1003
初期化ユーティリティ (dbinit), 959
データソースユーティリティ (dbdsn), 936
バックアップユーティリティ (dbbackup), 913
-m オプション
Broadcast Repeater ユーティリティ (dbns12), 921
ping ユーティリティ (dbping), 994
アンロードユーティリティ (dbunload), 1044
言語ユーティリティ (dblang), 983
証明書作成ユーティリティ (createcert), 924
初期化ユーティリティ (dbinit), 959
データベースオプション (dbeng12、dbsrv12), 311
データベースサーバーオプション (dbeng12、dbsrv12), 255
トランザクションログユーティリティ (dblog), 1041
ログ変換ユーティリティ (dbtran), 985
-nl オプション
アンロードユーティリティ (dbunload), 1044
-nogui オプション
Interactive SQL ユーティリティ (dbisql), 976
-no オプション
アンロードユーティリティ (dbunload), 1044
-nr オプション
サポートユーティリティ (dbsupport), 1036
-ns オプション
データソースユーティリティ (dbdsn), 937
-n オプション
アンロードユーティリティ (dbunload), 1044
サーバーオプション, 257
サーバー列挙ユーティリティ (dblocate), 999
初期化ユーティリティ (dbinit), 959
データベースオプション (dbeng12、dbsrv12), 312
データベース名の設定, 312
トランザクションログユーティリティ (dblog), 1041
バックアップユーティリティ (dbbackup), 913
ヒストグラムユーティリティ (dbhist), 955
ログ変換ユーティリティ (dbtran), 985
-od オプション
Linux サービスユーティリティ (dbsvc), 1008
-oe オプション
クワイエットモードでの操作, 160
ロギング起動エラー, 259
-onerror オプション
Interactive SQL ユーティリティ (dbisql), 976
-on オプション
データベースサーバーオプション (dbeng12、dbsrv12), 260
-op オプション
証明書ビューアーユーティリティ (viewcert), 934
-or オプション
データソースユーティリティ (dbdsn), 937
-os オプション
データベースサーバーオプション (dbeng12、dbsrv12), 261
-ot オプション
データベースサーバーオプション (dbeng12、dbsrv12), 262
-o オプション
Broadcast Repeater ユーティリティ (dbns12), 921
ping ユーティリティ (dbping), 994
SQL Anywhere スクリプト実行ユーティリティ (dbrunsql), 997
Windows サービスユーティリティ (dbsvc), 1019
アップグレードユーティリティ (dbupgrad), 1062
アンロードユーティリティ (dbunload), 1044
クワイエットモードでの操作, 160
検証ユーティリティ (dbvalid), 1064
サポートユーティリティ (dbsupport), 1027
サーバーライセンス取得ユーティリティ (dblic), 1003
サーバー列挙ユーティリティ (dblocate), 999
消去ユーティリティ (dberase), 951
情報ユーティリティ (dbinfo), 957
証明書ビューアーユーティリティ (viewcert), 934
初期化ユーティリティ (dbinit), 959

停止ユーティリティ (dbstop), 1024
データソースユーティリティ (dbdsn), 937
データベースサーバーオプション (dbeng12、dbsrv12), 258
トランザクションログユーティリティ (dblog), 1041
バックアップユーティリティ (dbbackup), 913
ログ変換ユーティリティ (dbtran), 985
-pc オプション
ping ユーティリティ (dbping), 994
サポートユーティリティ (dbsupport), 1028
データベースサーバーオプション (dbeng12、dbsrv12), 263
-pd オプション
ping ユーティリティ (dbping), 994
サポートユーティリティ (dbsupport), 1028
-pe オプション
データソースユーティリティ (dbdsn), 937
-port オプション
Interactive SQL ユーティリティ (dbisql), 976
SQL Anywhere コンソールユーティリティ (dbconsole), 1021
-pr オプション
Linux サービスユーティリティ (dbsvc), 1008
-ps オプション
ping ユーティリティ (dbping), 994
サポートユーティリティ (dbsupport), 1028
-pt オプション
データベースサーバーオプション (dbeng12、dbsrv12), 264
-p オプション
Broadcast Repeater ユーティリティ (dbns12), 921
Windows サービスユーティリティ (dbsvc), 1016
アンロードユーティリティ (dbunload), 1044
サーバー列挙ユーティリティ (dblocate), 999
証明書ビューアーユーティリティ (viewcert), 934
初期化ユーティリティ (dbinit), 959
データベースサーバーオプション (dbeng12、dbsrv12), 262
バックアップユーティリティ (dbbackup), 913
プロセス生成ユーティリティ (dbspawn), 1023
-qc オプション
SQL Anywhere スクリプト実行ユーティリティ (dbrunsql), 997
アンロードユーティリティ (dbunload), 1044
-qi オプション
Windows Mobile でサポート対象外, 438
クワイエットモードでの操作, 160
データベースサーバーオプション (dbeng12、dbsrv12), 264
-qn オプション
データベースサーバーオプション (dbeng12、dbsrv12), 265
-qp オプション
データベースサーバーオプション (dbeng12、dbsrv12), 266
-qr オプション
アンロードユーティリティ (dbunload), 1044
-qs オプション
クワイエットモードでの操作, 160
データベースサーバーオプション (dbeng12、dbsrv12), 266
-qw オプション
クワイエットモードでの操作, 160
データベースサーバーオプション (dbeng12、dbsrv12), 267
-q オプション
Broadcast Repeater ユーティリティ (dbns12), 921
dbisqlc ユーティリティ, 950
Interactive SQL ユーティリティ (dbisql), 976
Linux サービスユーティリティ (dbsvc), 1011
ping ユーティリティ (dbping), 994
SQL Anywhere スクリプト実行ユーティリティ (dbrunsql), 997
Windows サービスユーティリティ (dbsvc), 1019
アップグレードユーティリティ (dbupgrad), 1062
アンロードユーティリティ (dbunload), 1044
言語ユーティリティ (dblang), 983
検証ユーティリティ (dbvalid), 1064
サポートユーティリティ (dbsupport), 1027
サーバーライセンス取得ユーティリティ (dblic), 1003
サーバー列挙ユーティリティ (dblocate), 999
消去ユーティリティ (dberase), 951
情報ユーティリティ (dbinfo), 957
初期化ユーティリティ (dbinit), 959
停止ユーティリティ (dbstop), 1024
データソースユーティリティ (dbdsn), 937
トランザクションログユーティリティ (dblog), 1041

バックアップユーティリティ (dbbackup), 913
ファイル非表示 (dbfhide), 953
プロセス生成ユーティリティ (dbspawn), 1023
ログ変換ユーティリティ (dbtran), 985
-rd オプション
サポートユーティリティ (dbsupport), 1036
-rg オプション
Windows サービスユーティリティ (dbsvc), 1016
-rl オプション
Linux サービスユーティリティ (dbsvc), 1008
-ro オプション
証明書作成ユーティリティ (createcert), 924
-rr オプション
サポートユーティリティ (dbsupport), 1036
-rs オプション
Linux サービスユーティリティ (dbsvc), 1008
Windows サービスユーティリティ (dbsvc), 1016
-r オプション
アンロードユーティリティ (dbunload), 1044
サポートユーティリティ (dbsupport), 1036
証明書作成ユーティリティ (createcert), 924
データベースオプション (dbeng12、dbsrv12), 313
データベースサーバーオプション (dbeng12、dbsrv12), 267
トランザクションログユーティリティ (dblog), 1041
バックアップユーティリティ (dbbackup), 913
ログ変換ユーティリティ (dbtran), 985
-sa オプション
サポートユーティリティ (dbsupport), 1028
-sb オプション
データベースサーバーオプション (dbeng12、dbsrv12), 270
-scn オプション
証明書作成ユーティリティ (createcert), 924
-sc オプション
サポートユーティリティ (dbsupport), 1028
証明書作成ユーティリティ (createcert), 924
-sd オプション
Windows サービスユーティリティ (dbsvc), 1016
サポートユーティリティ (dbsupport), 1028
-sf オプション
データベースサーバーオプション (dbeng12、dbsrv12), 271
-sk オプション
データベースサーバーオプション (dbeng12、dbsrv12), 275
-sl オプション
証明書作成ユーティリティ (createcert), 924
-sm オプション
データベースオプション (dbsrv12), 314
ミラーデータベースへのアクセスに使用, 1165
-sn オプション
Windows サービスユーティリティ (dbsvc), 1016
データベースオプション (dbsrv12), 315
-sou オプション
証明書作成ユーティリティ (createcert), 924
-so オプション
証明書作成ユーティリティ (createcert), 924
-sr オプション
ログ変換ユーティリティ (dbtran), 985
-sst オプション
証明書作成ユーティリティ (createcert), 924
-ss オプション
サーバー列挙ユーティリティ (dblocate), 999
-status オプション
Linux サービスユーティリティ (dbsvc), 1008
-st オプション
ping ユーティリティ (dbping), 994
-su オプション
データベースサーバーオプション (dbeng12、dbsrv12), 276
ユーティリティデータベースへの接続, 137
-s オプション
Broadcast Repeater ユーティリティ (dbns12), 921
Linux サービスユーティリティ (dbsvc), 1008
ping ユーティリティ (dbping), 994
SQL Anywhere スクリプト実行ユーティリティ (dbrunsql), 997
Windows Mobile でサポート対象外, 438
Windows サービスユーティリティ (dbsvc), 1016
検証ユーティリティ (dbvalid), 1064
サーバー列挙ユーティリティ (dblocate), 999
証明書作成ユーティリティ (createcert), 924
初期化ユーティリティ (dbinit), 959
データベースサーバーオプション (dbeng12、dbsrv12), 268
バックアップユーティリティ (dbbackup), 913
ログ変換ユーティリティ (dbtran), 985

-ti オプション
データベースサーバーオプション (dbeng12、dbsrv12), 277
-tl オプション
データベースサーバーオプション (dbeng12、dbsrv12), 278
-tmf オプション
Windows Mobile でサポート対象外, 438
データベースサーバーオプション (dbeng12、dbsrv12), 279
-tmt オプション
Windows Mobile でサポート対象外, 438
データベースサーバーオプション (dbeng12、dbsrv12), 280
-tq time オプション
データベースサーバーオプション (dbeng12、dbsrv12), 280
-t オプション
Linux サービスユーティリティ (dbsvc), 1008
Windows サービスユーティリティ (dbsvc), 1013
アンロードユーティリティ (dbunload), 1044
検証ユーティリティ (dbvalid), 1064
証明書作成ユーティリティ (createcert), 924
初期化ユーティリティ (dbinit), 959
トランザクションログユーティリティ (dblog), 1041
バックアップユーティリティ (dbbackup), 913
ヒストグラムユーティリティ (dbhist), 955
ログ変換ユーティリティ (dbtran), 985
-ua オプション
Windows Mobile でサポート対象外, 438
データベースサーバーオプション (dbeng12、dbsrv12), 281
-uc オプション
Windows Mobile でサポート対象外, 438
データベースサーバーオプション (dbeng12、dbsrv12), 282
-ud オプション
Broadcast Repeater ユーティリティ (dbns12), 921
Windows Mobile でサポート対象外, 438
データベースサーバーオプション (dbeng12、dbsrv12), 282
-uf オプション
Windows Mobile でサポート対象外, 438
データベースサーバーオプション (dbeng12、dbsrv12), 283
-ui オプション
Broadcast Repeater ユーティリティ (dbns12), 921
Windows Mobile でサポート対象外, 438
データベースサーバーオプション (dbeng12、dbsrv12), 284
-ul オプション
Interactive SQL ユーティリティ (dbisql), 976
-um オプション
データベースサーバーオプション (dbeng12、dbsrv12), 284
-ut オプション
Windows Mobile でサポート対象外, 438
データベースサーバーオプション (dbeng12、dbsrv12), 285
-ux オプション
Broadcast Repeater ユーティリティ (dbns12), 921
Windows Mobile でサポート対象外, 438
データベースサーバーオプション (dbeng12、dbsrv12), 286
-u オプション
Linux サービスユーティリティ (dbsvc), 1007
Windows Mobile でサポート対象外, 438
Windows サービスユーティリティ (dbsvc), 1013
アンロードユーティリティ (dbunload), 1044
言語ユーティリティ (dblang), 983
サーバーライセンス取得ユーティリティ (dblic), 1003
情報ユーティリティ (dbinfo), 957
証明書作成ユーティリティ (createcert), 924
データベースサーバーオプション (dbeng12、dbsrv12), 281
ヒストグラムユーティリティ (dbhist), 955
ログ変換ユーティリティ (dbtran), 985
-version
Interactive SQL ユーティリティ (dbisql), 976
-vss オプション
データベースサーバーオプション (dbeng12/dbsrv12), 287
-v オプション
SQL Anywhere スクリプト実行ユーティリティ (dbrunsql), 997
アンロードユーティリティ (dbunload), 1044
サーバー列挙ユーティリティ (dblocate), 999
証明書作成ユーティリティ (createcert), 924
データソースユーティリティ (dbdsn), 937

データベースサーバーオプション (dbeng12、dbsrv12), 287
-wc オプション
サーバーオプション (dbeng12、dbsrv12), 288
データベースオプション (dbeng12、dbsrv12), 317
-wm オプション
ファイル非表示 (dbfhide), 953
-w オプション
Linux サービスユーティリティ (dbsvc), 1007
Windows サービスユーティリティ (dbsvc), 1013
データソースユーティリティ (dbdsn), 936
ファイル非表示 (dbfhide), 953
-xa オプション
データベースサーバーオプション (dbsrv12), 290
-xd オプション
データベースサーバーオプション (dbeng12、dbsrv12), 292
-xf オプション
データベースサーバーオプション (dbsrv12), 292
-xi オプション
アンロードユーティリティ (dbunload), 1044
-xm オプション
データベースサーバーオプション (dbeng12、dbsrv12), 293
-xo オプション
バックアップユーティリティ (dbbackup), 913
-xp オプション
Windows Mobile でサポート対象外, 438
データベースオプション (dbsrv12), 318
-xs オプション
クラウドサーバーで使用, 294
通信の保護, 1425
データベースサーバーオプション (dbeng12、dbsrv12), 294
-xx オプション
アンロードユーティリティ (dbunload), 1044
-x オプション
Broadcast Repeater ユーティリティ (dbns12), 921
dbisqlc ユーティリティ, 950
Interactive SQL ユーティリティ (dbisql), 976
Linux サービスユーティリティ (dbsvc), 1007
Windows サービスユーティリティ (dbsvc), 1013
証明書作成ユーティリティ (createcert), 924
停止ユーティリティ (dbstop), 1024
データベースサーバーオプション (dbeng12、dbsrv12), 289
トランザクションログユーティリティ (dblog), 1041
バックアップユーティリティ (dbbackup), 913
ログ変換ユーティリティ (dbtran), 985
-y オプション
Linux サービスユーティリティ (dbsvc), 1011
Windows サービスユーティリティ (dbsvc), 1019
アンロードユーティリティ (dbunload), 1044
消去ユーティリティ (dberase), 951
停止ユーティリティ (dbstop), 1024
データソースユーティリティ (dbdsn), 937
バックアップユーティリティ (dbbackup), 913
ログ変換ユーティリティ (dbtran), 985
-ze オプション
Windows Mobile でサポート対象外, 438
初期化ユーティリティ (dbinit), 959
データベースサーバーオプション (dbeng12、dbsrv12), 297
-zl オプション
データベースサーバーオプション (dbeng12、dbsrv12), 297
-zn オプション
初期化ユーティリティ (dbinit), 959
データベースサーバーオプション (dbeng12、dbsrv12), 298
-zoc オプション
データベースサーバーオプション (dbeng12、dbsrv12), 300
-zo オプション
データベースサーバーオプション (dbeng12、dbsrv12), 299
-zp オプション
データベースサーバーオプション (dbeng12、dbsrv12), 301
-zr オプション
データベースサーバーオプション (dbeng12、dbsrv12), 301
-zs オプション
データベースサーバーオプション (dbeng12、dbsrv12), 303
-zt オプション
データベースサーバーオプション (dbeng12、dbsrv12), 304

-z オプション
Broadcast Repeater ユーティリティ (dbns12), 921
ping ユーティリティ (dbping), 994
初期化ユーティリティ (dbinit), 959
データベースサーバーオプション (dbeng12、dbsrv12), 296
トランザクションログユーティリティ (dblog), 1041
ネットワーク通信問題のデバッグ, 1211
ログ変換ユーティリティ (dbtran), 985

## A

AccentSensitive プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
ActiveReq 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
ActiveReq プロパティ
サーバープロパティの説明, 754
ActiveSync
Vista, 188
Windows Server 2008, 188
必要なバージョン, 404
ActiveSync プロバイダーインストールユーティリティ (mlasinst)
Vista での権限の昇格の必要性, 188
ActiveSync プロバイダーインストールユーティリティ (mlasinst)
Windows 7 での権限の昇格の必要性, 188
Windows Server 2008 での権限の昇格の必要性, 188
Address Windowing Extensions
キャッシュサイズの制限, 217
admin ユーザー
モニター、説明, 1297
ADO.NET Sample
使用, 408
AES_FIPS 暗号化アルゴリズム
-fips サーバーオプション, 230
アンロードユーティリティ (dbunload), 1044
初期化ユーティリティ (dbinit), 959
AES256_FIPS 暗号化アルゴリズム
-fips サーバーオプション, 230
アンロードユーティリティ (dbunload), 1044
初期化ユーティリティ (dbinit), 959
AES256 暗号化アルゴリズム
アンロードユーティリティ (dbunload), 1044
初期化ユーティリティ (dbinit), 959
AES 暗号化アルゴリズム
アンロードユーティリティ (dbunload), 1044
初期化ユーティリティ (dbinit), 959
説明, 1397
Agent テーブル
SQL Anywhere MIB, 1334
AIX
IPv6 サポート, 100
LDAP サーバーの使用, 103
LIBPATH 環境変数, 446
Alias プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
allow_nulls_by_default オプション
ASE 互換性オプション, 585
SQL Anywhere SNMP Extension Agent OID, 1362
Sybase Open Client, 1443
Transact-SQL 互換性オプション, 585
説明, 589
allow_nulls_by_default プロパティ
接続プロパティの説明, 713
allow_read_client_file オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 590
allow_read_client_file プロパティ
接続プロパティの説明, 713
allow_snapshot_isolation オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 590
allow_snapshot_isolation プロパティ
接続プロパティの説明, 713
allow_write_client_file オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 592
allow_write_client_file プロパティ
接続プロパティの説明, 713
ALL パーミッション
説明, 520
付与, 526

ALTER DATABASE 文
Windows Mobile の制限, 436
プライマリサーバーのシャットダウン, 1166
ミラーリングシステムでのフェールオーバーの強制, 1168
ALTER LOGIN POLICY 文
ログインポリシーの変更, 559
AlternateMirrorServerName プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
AlternateServerName プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
ALTER USER 文
既存のユーザーへのログインポリシーの割り当て, 558
パスワード, 515
ALTER パーミッション
説明, 520
付与, 526
ANSI
cooperative_commits オプション, 611
delayed_commits オプション, 620
カーソル, 593
更新パーミッション, 594
コードページの説明, 476
削除パーミッション, 594
準拠, 679
データベースの互換性オプション, 584
変数の性質, 592
ansi_blanks オプション
ASE 互換性オプション, 585
SQL Anywhere SNMP Extension Agent OID, 1362
Sybase Open Client, 1443
Transact-SQL 互換性, 585
説明, 592
ansi_blanks プロパティ
接続プロパティの説明, 713
ansi_close_cursors_on_rollback オプション
SQL Anywhere SNMP Extension Agent OID, 1362
Transact-SQL 互換性オプション, 585
説明, 593
ansi_close_cursors_on_rollback プロパティ
接続プロパティの説明, 713
ansi_permissions オプション
ASA SNMP Extension Agent OID, 1362
Transact-SQL 互換性オプション, 585
説明, 594
ansi_permissions プロパティ
接続プロパティの説明, 713
ansi_substring オプション
ASE 互換性オプション, 585
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 595
ansi_substring プロパティ
接続プロパティの説明, 713
ansi_update_constraints オプション
SQL Anywhere SNMP Extension Agent OID, 1362
Transact-SQL 互換性オプション, 585
説明, 596
ansi_update_constraints プロパティ
接続プロパティの説明, 713
ansinull オプション
ASE 互換性オプション, 585
SQL Anywhere SNMP Extension Agent OID, 1362
Sybase Open Client, 1443
Transact-SQL 互換性オプション, 585
説明, 597
ansinull プロパティ
接続プロパティの説明, 713
API
SQL Anywhere からの接続, 49
AppInfo 接続パラメーター
説明, 321
AppInfo プロパティ
接続プロパティの説明, 713
ApproximateCPUTime プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341, 1355
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
APP 接続パラメーター
説明, 321
ArbiterState プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
ASCII

ServerName (Server) 接続パラメーターの制限, 365
文字セット, 474
ASE
代替の文字セットエンコードラベル, 476
ASTART 接続パラメーター
説明, 324
ASTOP 接続パラメーター
説明, 324
auditing_options オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 599
auditing_options プロパティ
接続プロパティの説明, 713
AuditingTypes プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
auditing オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 598
auditing プロパティ
接続プロパティの説明, 713
authdn パラメーター
LDAP, 105
authenticate.sql
認証データベースのアップグレード, 186
Authenticated プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
接続プロパティの説明, 713
データベースプロパティの説明, 782
AuthType プロパティ
接続プロパティの説明, 713
auto_add_fan_out オプション
使用, 1200
auto_commit オプション
Interactive SQL 設定, 884
説明, 885
auto_refetch オプション
Interactive SQL 設定, 884
説明, 886
automatic_timestamp オプション
Transact-SQL 互換性オプション, 585
AutoMultiProgrammingLevelStatistics プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
AutoMultiProgrammingLevel プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
AutoStart 接続パラメーター
説明, 324
AutoStop 接続パラメーター
説明, 324
AvailIO 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
AvailIO プロパティ
サーバープロパティの説明, 754
AWE キャッシュ
-cm サーバーオプション, 212
-cw サーバーオプション, 217
Vista での SQL Anywhere の実行, 188
Windows 7 での SQL Anywhere の実行, 188
Windows Server 2008 での SQL Anywhere の実行, 188

## B

background_priority オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明 [旧式], 599
background_priority プロパティ
接続プロパティの説明, 713
backup.syb
ロケーションの取得, 450
BackupEnd システムイベント
説明, 1127
BACKUP 文
Windows Mobile の制限, 436
アーカイブバックアップの作成, 1081
イメージバックアップの作成, 1081
basedn パラメーター
LDAP, 105
BCAST プロトコルオプション
IPv6 アドレスの使用, 101
説明, 372
bell オプション
Interactive SQL 設定, 884
説明, 886

BlankPadding プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
BLISTENER プロトコルオプション
説明, 373
BLOB
共有, 116
説明, 116
データベースへの格納, 116
blob_threshold オプション
SQL Remote 説明, 600
SQL Remote レプリケーションオプション, 588
BlockedOn プロパティ
接続プロパティの説明, 713
blocking
blocking_others_timeout オプション, 601
blocking_others_timeout オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 601
blocking_others_timeout プロパティ
接続プロパティの説明, 713
blocking_timeout オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 602
blocking_timeout プロパティ
接続プロパティの説明, 713
blocking オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 601
blocking プロパティ
接続プロパティの説明, 713
BroadcastListener プロトコルオプション
説明, 373
Broadcast Repeater ユーティリティ (dbns12)
構文, 921
使用, 82
Broadcast プロトコルオプション
説明, 372
BuildChange プロパティ
サーバープロパティの説明, 754
BuildClient プロパティ
サーバープロパティの説明, 754
BuildProduction プロパティ
サーバープロパティの説明, 754
BuildReproducible プロパティ
サーバープロパティの説明, 754
BytesReceivedUncomp 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
BytesReceivedUncomp プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
BytesReceived 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
BytesReceived プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
BytesSentUncomp 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
BytesSentUncomp プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
BytesSent 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
BytesSent プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782

## C

CacheAllocated 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
CacheAllocated プロパティ
サーバープロパティの説明, 754
CacheFileDirty 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
CacheFileDirty プロパティ
サーバープロパティの説明, 754
CacheFile 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
CacheFile プロパティ
サーバープロパティの説明, 754

CacheFree 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
CacheFree プロパティ
サーバープロパティの説明, 754
CacheHits 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
CacheHits プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
CachePanics 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
CachePanics プロパティ
サーバープロパティの説明, 754
CachePinned 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
CachePinned プロパティ
サーバープロパティの説明, 754
CacheReadIndInt 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
CacheReadIndInt プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
CacheReadIndLeaf 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
CacheReadIndLeaf プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
CacheReadTable 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
CacheReadTable プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
CacheReadWorkTable 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
CacheReadWorkTable プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
CacheRead 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
CacheRead プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
CacheReplacements 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
CacheReplacements プロパティ
サーバープロパティの説明, 754
CacheScavenges 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
CacheScavenges プロパティ
サーバープロパティの説明, 754
CacheScavengeVisited 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
CacheScavengeVisited プロパティ
サーバープロパティの説明, 754
CacheSizingStatistics プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
Capabilities プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
CarverHeapPages 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
CarverHeapPages プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
CaseSensitive プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
CatalogCollation プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
CBSIZE 接続パラメーター
説明, 326
CBSIZE 通信パラメーター
TCP/IP, 95

CD-ROM
配備, 267
Certicom
クライアント／サーバー通信の暗号化, 223
certificate_company プロトコルオプション
説明, 374
certificate_name プロトコルオプション
説明, 375
certificate_unit プロトコルオプション
説明, 376
chained オプション
ASE 互換性オプション, 585
SQL Anywhere SNMP Extension Agent OID, 1362
Sybase Open Client, 1443
Transact-SQL 互換性オプション, 585
説明, 603
chained プロパティ
接続プロパティの説明, 713
char_charset エイリアス
説明, 477
CharSet 接続パラメーター
説明, 325
CharSet プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341, 1355
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
CHAR 照合
説明, 491
CHAR データ型
新規データベースでの照合順, 959
新規データベース用のエンコード, 959
ホスト変数, 592
checkpoint_time オプション
SQL Anywhere SNMP Extension Agent OID, 1362
使用, 1114
説明, 603
checkpoint_time プロパティ
接続プロパティの説明, 713
CheckpointLogCommitToDisk 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
CheckpointLogCommitToDisk プロパティ
データベースプロパティの説明, 782
CheckpointLogPagesInUse 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
CheckpointLogPagesInUse プロパティ
データベースプロパティの説明, 782
CheckpointLogPagesRelocated 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
CheckpointLogPagesRelocated プロパティ
データベースプロパティの説明, 782
CheckpointLogPagesWritten 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
CheckpointLogPagesWritten プロパティ
データベースプロパティの説明, 782
CheckpointLogSavePreimage 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
CheckpointLogSavePreimage プロパティ
データベースプロパティの説明, 782
CheckpointLogSize 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
CheckpointLogSize プロパティ
データベースプロパティの説明, 782
CheckpointLogWrites 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
CheckpointLogWrites プロパティ
データベースプロパティの説明, 782
CheckpointUrgency 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
CheckpointUrgency プロパティ
データベースプロパティの説明, 782
Checksum プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
child_creation オプション
使用, 1197
ChkptFlush 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
ChkptFlush プロパティ
データベースプロパティの説明, 782
ChkptPage 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350

ChkptPage プロパティ
データベースプロパティの説明, 782
Chkpt 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
Chkpt プロパティ
データベースプロパティの説明, 782
cis_option オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 604
cis_option プロパティ
接続プロパティの説明, 713
cis_rowset_size オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 605
cis_rowset_size プロパティ
接続プロパティの説明, 713
CleanablePagesAdded プロパティ
データベースプロパティの説明, 782
CleanablePagesCleaned プロパティ
データベースプロパティの説明, 782
CleanableRowsAdded プロパティ
データベースプロパティの説明, 782
CleanableRowsCleaned プロパティ
データベースプロパティの説明, 782
ClientLibrary プロパティ
接続プロパティの説明, 713
ClientNodeAddress プロパティ
接続プロパティの説明, 713
ClientPort プロトコルオプション
説明, 377
ClientPort プロパティ
接続プロパティの説明, 713
ClientStmtCacheHits 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
ClientStmtCacheHits プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
ClientStmtCacheMisses 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
ClientStmtCacheMisses プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
close_on_endtrans オプション
SQL Anywhere SNMP Extension Agent OID, 1362
Sybase Open Client, 1443
Transact-SQL 互換性オプション, 585
説明, 605
close_on_endtrans プロパティ
接続プロパティの説明, 713
Collation プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
collect_statistics_on_dml_updates オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 606
collect_statistics_on_dml_updates プロパティ
接続プロパティの説明, 713
CollectStatistics プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
command_delimiter オプション
Interactive SQL 設定, 884
説明, 887
CommandLine プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
CommBufferSize 接続パラメーター
TCP/IP, 95
説明, 326
commit_on_exit オプション
Interactive SQL 設定, 884
説明, 888
CommitFile 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
CommitFile プロパティ
データベースプロパティの説明, 782
Commit 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
Commit プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782

COMMIT 文
auto_commit オプション, 885
Interactive SQL のショートカット, 851
CommLinks 接続パラメーター
IPv6 アドレス, 100
オプション, 97
カッコ, 478
説明, 327
CommLink プロパティ
接続プロパティの説明, 713
CommNetworkLink プロパティ
接続プロパティの説明, 713
CommProtocol プロパティ
接続プロパティの説明, 713
CompactPlatformVer プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
CompanyName プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
CompressionThreshold 接続パラメーター
説明, 330
compression オプション
SQL Remote レプリケーションオプション, 588
説明, 607
Compression プロパティ
接続プロパティの説明, 713
Compress 接続パラメーター
説明, 329
COMPTH 接続パラメーター
説明, 330
COMP 接続パラメーター
説明, 329
conn_auditing オプション
SQL Anywhere SNMP Extension Agent OID, 1362
使用, 1390
説明, 607
conn_auditing プロパティ
接続プロパティの説明, 713
ConnCount 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
ConnCount プロパティ
データベースプロパティの説明, 782
ConnectedTime プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341, 1355
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
ConnectFailed システムイベント
login_procedure オプションの例, 638
説明, 1127
例, 1129
connection_authentication オプション
SQL Anywhere SNMP Extension Agent OID, 1362
使用, 184
説明, 608
connection_authentication プロパティ
接続プロパティの説明, 713
CONNECTION_PROPERTY 関数
オプション値の取得, 571
接続プロパティのアルファベット順リスト, 713
ConnectionName 接続パラメーター
説明, 332
ConnectionPool 接続パラメーター
Windows Mobile でサポート対象外, 435
説明, 332
Connect システムイベント
説明, 1127
ConnPoolCachedCount 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
ConnPoolCachedCount プロパティ
データベースプロパティの説明, 782
ConnPoolHits 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
ConnPoolHits プロパティ
データベースプロパティの説明, 782
ConnPoolMisses 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
ConnPoolMisses プロパティ
データベースプロパティの説明, 782
ConnsDisabled プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341, 1355
サーバープロパティの説明, 754
データベースプロパティの説明, 782
ConsoleLogFile プロパティ

サーバープロパティの説明, 754
ConsoleLogFile プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
ConsoleLogMaxSize プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
continue_after_raiserror オプション
ASE 互換性オプション, 585
SQL Anywhere SNMP Extension Agent OID, 1362
Transact-SQL 互換性オプション, 585
説明, 609
continue_after_raiserror プロパティ
接続プロパティの説明, 713
conversion_error オプション
SQL Anywhere SNMP Extension Agent OID, 1362
Transact-SQL 互換性オプション, 585
説明, 610
conversion_error プロパティ
接続プロパティの説明, 713
CON 接続パラメーター
説明, 332
cooperative_commit_timeout オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 610
cooperative_commit_timeout プロパティ
接続プロパティの説明, 713
cooperative_commits オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 611
cooperative_commits プロパティ
接続プロパティの説明, 713
CPOOL 接続パラメーター
Windows Mobile でサポート対象外, 435
説明, 332
CPORT プロトコルオプション
説明, 377
CPU
-gt サーバーオプション, 244
使用する数, 158
createcert ユーティリティ
構文, 924
使用法, 1414
CREATE DATABASE 文
Windows Mobile の制限, 436
Windows Mobile 用データベースの作成, 421
使用, 118
パーミッション, 159
ファイル管理文パーミッション, 561
ユーティリティデータベース, 135
CREATE DBSPACE 文
使用, 124
CREATE DECRYPTED DATABASE 文
使用, 1401
CREATE DECRYPTED FILE 文
テクニカルサポートのためのデータベースの復号化, 1401
CREATE ENCRYPTED DATABASE 文
CREATE ENCRYPTED FILE 文との比較, 1400
使用, 1400
CREATE ENCRYPTED FILE 文
CREATE ENCRYPTED DATABASE 文との比較, 1400
テクニカルサポートのためのデータベースの暗号化, 1400
CREATE EVENT 文
Windows Mobile の制限, 436
CREATE EXISTING TABLE 文
Windows Mobile でサポート対象外, 436
CREATE EXTERNLOGIN 文
Windows Mobile でサポート対象外, 436
CREATE FUNCTION 文
Windows Mobile の制限, 436
createkey ユーティリティ
構文, 981
CREATE LOGIN POLICY 文
新しいログインポリシーの作成, 557
パスワードセキュリティの強化, 1383
CREATE MIRROR SERVER 文
データベースミラーリングの設定, 1160
CREATE ON パーミッション
説明, 520
CREATE SERVER 文
Windows Mobile でサポート対象外, 436
CREATE TABLE 文
Windows Mobile の制限, 436
CREATE USER 文
使用, 514
新規ユーザーのデフォルト, 513
パスワードを持たない, 550

ユーザーの作成とログインポリシーの割り当て, 558
CREATE パーミッション
DB 領域, 122
CSFC5KTNAME 環境変数
Kerberos, 63
CSV ファイル
モニター, 1252
CS 接続パラメーター
説明, 325
CurrentCacheSize 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
CurrentCacheSize プロパティ
サーバープロパティの説明, 754
CurrentLineNumber プロパティ
接続プロパティの説明, 713
CurrentMultiProgrammingLevel プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
CurrentProcedure プロパティ
接続プロパティの説明, 713
CurrentRedoPos プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
CURRENT USER
環境設定, 464
CurrIO 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
CurrIO プロパティ
データベースプロパティの説明, 782
CurrRead 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
CurrRead プロパティ
データベースプロパティの説明, 782
CurrWrite 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
CurrWrite プロパティ
データベースプロパティの説明, 782
CursorOpen 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
CursorOpen プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
Cursor 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
Cursor プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
CyberSafe Kerberos クライアント
UNIX サポート, 63
Windows サポート, 63

## D

DAC
ネストされたビューとテーブルの規則, 566
database_authentication オプション
SQL Anywhere SNMP Extension Agent OID, 1362
使用, 183
説明, 612
database_authentication プロパティ
接続プロパティの説明, 713
DatabaseCleaner プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
DatabaseFile 接続パラメーター
組み込みデータベース, 28
説明, 333
DatabaseKey 接続パラメーター
説明, 335
DatabaseName 接続パラメーター
説明, 336
DatabaseName プロトコルオプション
説明, 379
DatabaseStart システムイベント
説明, 1127
DatabaseSwitches 接続パラメーター
説明, 337
DataSourceName 接続パラメーター
Windows Mobile, 47
説明, 338
date_format オプション
SQL Anywhere SNMP Extension Agent OID, 1362

Sybase Open Client, 1443
Transact-SQL 互換性オプション, 585
説明, 613
date_format プロパティ
接続プロパティの説明, 713
date_order オプション
SQL Anywhere SNMP Extension Agent OID, 1362
Sybase Open Client, 1443
Transact-SQL 互換性オプション, 585
説明, 615
date_order プロパティ
接続プロパティの説明, 713
db_charset
説明, 476
DB_PROPERTY 関数
データベースプロパティのアルファベット順リスト, 781
DBA
継承不可能な権限, 538
権限, 538
権限の制限, 1380
新規データベースでの指定, 959
認証, 538
ログインポリシー, 538
DBA 権限
付与, 542
dbbackup ユーティリティ
エラーの受信, 913
クライアント側バックアップ, 1085
構文, 913
終了コード, 920
フルバックアップ, 1074
ライブバックアップ, 1075
dbcapi.dll
クライアント API と文字セット変換, 480
dbconsole ユーティリティ
Mac OS X のハードウェア要件, 906
起動, 905
構文, 1021
使用, 905
ソフトウェア更新, 907
ユーザーとの接続の切断, 543
dbctrs12.dll
Vista での権限の昇格の必要性, 188
Windows 7 での権限の昇格の必要性, 188
Windows Server 2008 での権限の昇格の必要性, 188
DBDiskSpace システムイベント
説明, 1127
例, 1129
dbdsn ユーティリティ
Vista での権限の昇格の必要性, 188
Windows 7 での権限の昇格の必要性, 188
Windows Server 2008 での権限の昇格の必要性, 188
構文, 936
システム情報ファイル, 940
終了コード, 940
使用, 42
dbelevate12.exe
Vista での権限の昇格の必要性, 188
Vista への配備に含めることが必要, 188
Windows 7 での権限の昇格の必要性, 188
Windows 7 への配備に含めることが必要, 188
Windows Server 2008 での権限の昇格の必要性, 188
Windows Server 2008 への配備に含めることが必要, 188
dbeng12
Linux サービスとして実行, 1007
Windows サービスとして実行, 1013
構文, 189
コマンドライン, 189
パーソナルデータベースサーバー, 141
ライセンス, 1003
dberase ユーティリティ
構文, 951
終了コード, 953
使用, 141
DBExpress
クライアント API と文字セット変換, 480
dbfhide ユーティリティ
構文, 953
DBFileFragments プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
DBF 接続パラメーター
組み込みデータベース, 28
説明, 333
dbhist ユーティリティ
構文, 955
終了コード, 956
dbicu12.dll
Windows Mobile での ICU の回避, 419

Windows Mobile でのデータベースのアンロード, 424
Windows Mobile 用データベースの作成, 419
dbicudt12.dll
Windows Mobile での ICU の回避, 419
Windows Mobile でのデータベースのアンロード, 424
Windows Mobile 用データベースの作成, 419
dbinfo ユーティリティ
構文, 957
終了コード, 958
ディスク上のテーブルのサイズの判断, 958
dbinit ユーティリティ
Windows Mobile 用データベースの作成, 421
構文, 958
終了コード, 975
使用, 119
dbinit を使用したデータベースの初期化
説明, 958
dbisql.com
説明, 980
dbisql.exe
高速ランチャーオプション, 903
説明, 980
dbisqlc ユーティリティ
構文, 949
サポートされるプラットフォーム, 950
dbisql ユーティリティ
構文, 975
サポートされるプラットフォーム, 980
終了コード, 981
説明, 840
DBKEY 接続パラメーター
説明, 335
dblang ユーティリティ
高速ランチャーが有効なときに使用, 904
構文, 983
終了コード, 984
説明, 983
DBLauncher
Mac OS X でのデータベースサーバーの起動, 9
dblgen12.res
検索, 461
DBLIB
クライアント API と文字セット変換, 480
dblic ユーティリティ
Vista での権限の昇格の必要性, 188
Windows 7 での権限の昇格の必要性, 188
Windows Server 2008 での権限の昇格の必要性, 188
構文, 1003
終了コード, 1006
dblocate ユーティリティ
構文, 999
終了コード, 1002
dblog ユーティリティ
監査, 1395
構文, 1041
コマンドライン, 1041
終了コード, 1044
トランザクションログミラー, 130
dblsn ユーティリティ
Windows サービスとして実行, 1013
dbmlsync ユーティリティ
Linux サービスとして実行, 1007
TLS, 1431
Windows サービスとして実行, 1013
DBNS
定義, 82
dbns12 ユーティリティ
Linux サービスとして実行, 1007
Windows サービスとして実行, 1013
構文, 921
使用, 82
DBNumber プロパティ
接続プロパティの説明, 713
DBN 接続パラメーター
説明, 336
DBN プロトコルオプション
説明, 379
dbodbc12.dll
Vista での権限の昇格の必要性, 188
Windows 7 での権限の昇格の必要性, 188
Windows Server 2008 での権限の昇格の必要性, 188
dboledb12.dll
Vista での権限の昇格の必要性, 188
Windows 7 での権限の昇格の必要性, 188
Windows Server 2008 での権限の昇格の必要性, 188
dboledba12.dll
Vista での権限の昇格の必要性, 188
Windows 7 での権限の昇格の必要性, 188
Windows Server 2008 での権限の昇格の必要性, 188
dbo ユーザー

システムオブジェクトとアンロードユーティリティ, 1057
説明, 551
dbping_r ユーティリティ
UNIX での使用, 994
dbping ユーティリティ
構文, 994
終了コード, 997
使用, 93
dbremote ユーティリティ
Linux サービスとして実行, 1007
Windows サービスとして実行, 1013
dbrunsql ユーティリティ
構文, 997
dbsnmp12.dll
説明, 1320
dbspawn ユーティリティ
構文, 1022
終了コード, 1024
dbsrv12
Linux サービスとして実行, 1007
Windows Mobile, 426
Windows サービスとして実行, 1013
構文, 189
コマンドライン, 189
トランスポートレイヤーセキュリティ, 1420
ネットワークデータベースサーバー, 141
ライセンス, 1003
dbstats ユーティリティ
構文, 991
終了コード, 993
dbstop ユーティリティ
SQLCONNECT の使用, 1026
構文, 1024
終了コード, 1026
使用, 146
パーミッション, 235
dbsupport.ini
説明, 1032
dbsupport ユーティリティ
SADIAGDIR 環境変数, 449
構文, 1027
使用, 1210
dbsvc ユーティリティ
Linux オプション, 1007
Linux 構文, 1007
Vista での権限の昇格の必要性, 188
Windows 7 での権限の昇格の必要性, 188
Windows Server 2008 での権限の昇格の必要性, 188
Windows オプション, 1013
Windows 構文, 1013
終了コード, 1020
DBS 接続パラメーター
説明, 337
DBTools インターフェイス
クライアント API と文字セット変換, 480
dbtran ユーティリティ
監査, 1395
監査情報の取り出し, 1391
構文, 985
コマンドライン, 985
コミットされない変更, 1090
終了コード, 991
使用, 1094
トランザクションログ, 1090
dbunload ユーティリティ
DB 領域ファイル名, 1044
構文, 1044
終了コード, 1058
dbupgrad ユーティリティ
構文, 1061
終了コード, 1064
dbvalid ユーティリティ
構文, 1064
終了コード, 1068
使用, 1074
dbversion ユーティリティ
構文, 1068
dbvss12.exe
SQL Anywhere VSS ライター, 1084
-vss dbeng12/dbsrv12 サーバーオプション, 287
Windows サービスとして実行, 1013
dbxtract ユーティリティ
Windows Mobile でサポート対象外, 441
DB 領域
CREATE ON パーミッション, 520
default_dbspace オプション, 617
-dt サーバーオプションによるロケーションの指定, 308
アンロード中のファイル名の変更, 1044
削除, 126
作成, 123
事前定義, 120
制限, 811
説明, 121

大容量データベース用の使用, 121
パーミッション, 122, 528
変更, 125
DB 領域作成ウィザード
使用, 124
Deadlock システムイベント
説明, 1127
debug_messages オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 616
debug_messages プロパティ
接続プロパティの説明, 713
DebuggingInformation プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
DECRYPT 関数
カラムの復号化に使用, 1403
dedicated_task オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 617
dedicated_task プロパティ
接続プロパティの説明, 713
default_dbspace オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 617
データベースオブジェクトの場所の指定, 121
default_dbspace プロパティ
接続プロパティの説明, 713
default_isql_encoding オプション
Interactive SQL 設定, 884
構文, 888
default_timestamp_increment オプション
Mobile Link 同期での使用, 619
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 618
default_timestamp_increment プロパティ
接続プロパティの説明, 713
DefaultCollation プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
説明, 135
DefaultNcharCollation プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
delayed_commit_timeout オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 619
delayed_commit_timeout プロパティ
接続プロパティの説明, 713
delayed_commits オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 620
delayed_commits プロパティ
接続プロパティの説明, 713
delete_old_logs オプション
SQL Remote 説明, 621
SQL Remote レプリケーションと同期オプション, 588
トランケーションオフセットのリセット, 1041
トランザクションログオプション, 1041
DELETE パーミッション
説明, 520
付与, 526
DELETE 文
Interactive SQL での生成, 859
Delphi
バイナリカラム, 650
Delphi 接続パラメーター
ODBC 接続パラメーターの説明, 942
demo.db ファイル
パーソナルサーバーサンプルの実行, 1
DER コード化 PKI オブジェクト
表示, 934
DescribeCursor 接続パラメーター
ODBC 接続パラメーターの説明, 942
Description 接続パラメーター
ODBC 接続パラメーターの説明, 942
DIAGNOSTICS グループ
説明, 551
DisableMultiRowFetch 接続パラメーター
説明, 339
Disconnect システムイベント
説明, 1127
DISCONNECT 文
使用, 543
DiskReadHintPages プロパティ
接続プロパティの説明, 713

データベースプロパティの説明, 782
DiskReadHintScatterLimit プロパティ
サーバープロパティの説明, 754
DiskReadHint プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
DiskReadIndInt 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
DiskReadIndInt プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
DiskReadIndLeaf 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
DiskReadIndLeaf プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
DiskReadTable 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
DiskReadTable プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
DiskReadWorkTable 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
DiskReadWorkTable プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
DiskRead 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
DiskRead プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
DiskRetryReadScatter プロパティ
サーバープロパティの説明, 754
データベースプロパティの説明, 782
DiskRetryRead プロパティ
サーバープロパティの説明, 754
DiskRetryWrite プロパティ
サーバープロパティの説明, 754
DiskSyncRead プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
DiskSyncWrite プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
DiskWaitRead プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
DiskWaitWrite プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
DiskWriteHintPages プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
DiskWriteHint プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
DiskWrite 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
DiskWrite プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
divide_by_zero_error オプション
SQL Anywhere SNMP Extension Agent OID, 1362
Transact-SQL 互換性オプション, 585
説明, 622
divide_by_zero_error プロパティ
接続プロパティの説明, 713
DLL
ロケーション, 460
DMRF 接続パラメーター
説明, 339
DoBroadcast プロトコルオプション
説明, 379
データベースミラーリング接続, 1163
DOBROAD プロトコルオプション
説明, 379
データベースミラーリング接続, 1163
DriveBus プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
DriveModel プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
Driver 接続パラメーター
ODBC 接続パラメーターの説明, 942
DriveType プロパティ

SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
DROP CONNECTION 文
使用, 543
DROP DATABASE 文
Windows Mobile でサポート対象外, 436
使用, 140
DROP LOGIN POLICY 文
ログインポリシーの削除, 560
DROP MIRROR SERVER 文
コピーノードの削除, 1202
ミラーサーバーの削除, 1169
DROP SERVER 文
Windows Mobile でサポート対象外, 436
DSEdit ユーティリティ
SQL Anywhere には含まれていない, 1437
エントリ, 1440
起動, 1438
使用, 1438
説明, 1437
DSN 接続パラメーター
Windows Mobile, 47
説明, 36, 338
DSN 接続パラメーター
接続, 18
DUMMY
システムテーブルに対するパーミッション, 519
DYLD_LIBRARY_PATH 環境変数
説明, 445

## E

ECC
サポート, 1411
ECC オプション
dbeng12 -ec, 223
dbsrv12 -ec, 223
ECC 証明書
作成, 924
表示, 934
echo オプション
Interactive SQL 設定, 884
説明, 889
Elevate 接続パラメーター
説明, 340
Embedded SQL
インターフェイスライブラリ, 77
接続, 49
接続のパフォーマンス, 93
接続のパフォーマンスのテスト, 93
EMOTE_IDLE_TIMEOUT オプション
説明, 670
EncryptedPassword 接続パラメーター
説明, 341
EncryptionScope プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
Encryption 接続パラメーター
説明, 342
Encryption プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
ENCRYPT 関数
カラムの暗号化に使用, 1403
ENC 接続パラメーター
クライアント／サーバー通信の保護, 1422
説明, 342
EngineName 接続パラメーター
廃止されたため代わりに ServerName を使用, 365
ENG 接続パラメーター
廃止されたため代わりに Server を使用, 365
ENP 接続パラメーター
説明, 341
ERRORLEVEL 環境変数
Interactive SQL リターンコード, 976
ER (実体関連) タブ
使用, 835
ER 図
SQL Anywhere 12 プラグインからの表示, 835
[ER 図] タブ
説明, 835
escape_character オプション
ASE 互換性オプション, 585
SQL Anywhere SNMP Extension Agent OID, 1362
Sybase Open Client, 1443
Transact-SQL 互換性オプション, 585
説明, 622
escape_character プロパティ
接続プロパティの説明, 713
Escape 接続パラメーター

ODBC 接続パラメーターの説明, 942
[Esc] キー
Interactive SQL, 878
ESQL Sample
使用, 409
Ethernet
説明, 95
EventName プロパティ
接続プロパティの説明, 713
EventTypeDesc プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
EventTypeName プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
ExchangeTasksCompleted プロパティ
サーバープロパティの説明, 754
ExchangeTasks 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
ExchangeTasks プロパティ
サーバープロパティの説明, 754
exclude_operators オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 623
exclude_operators プロパティ
接続プロパティの説明, 713
ExprCacheAbandons プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
ExprCacheDropsToReadOnly プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
ExprCacheEvicts プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
ExprCacheHits プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
ExprCacheInserts プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
ExprCacheLookups プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
ExprCacheResumesOfReadWrite プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
ExprCacheStarts プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
ExtendDB 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
ExtendDB プロパティ
データベースプロパティの説明, 782
extended_join_syntax オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 623
extended_join_syntax プロパティ
接続プロパティの説明, 713
ExtendTempWrite 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
ExtendTempWrite プロパティ
データベースプロパティの説明, 782
EXTENV_MAIN ユーザー
説明, 551
EXTENV_WORKER ユーザー
説明, 551
external_remote_options オプション
SQL Remote のオプション, 623

## F

FileDataSourceName 接続パラメーター
Windows Mobile, 47
説明, 344
ファイルデータソースの参照, 36
FILEDSN 接続パラメーター
Windows Mobile, 47
説明, 344
ファイルデータソースの参照, 36
FILEDSN 接続パラメーター
接続, 18
説明, 45
FileSize プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
File プロパティ

SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
Finder
環境変数の設定, 444
FIPS
dbeng12 -ec, 223
dbeng12 -fips, 230
dbinit -ea, 959
dbsrv12 -ec, 223
dbsrv12 -fips, 230
FIPS プロトコルオプション, 385
SQL_FLAGGER_ERROR オプション, 679
Web サービス, 294
サポート, 1411
説明, 1412
データベースファイルの暗号化, 1044
FIPS 140-2 認定
説明, 1412
FipsMode プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
FIPS オプション
AES_FIPS 暗号化アルゴリズム, 230
AES256_FIPS 暗号化アルゴリズム, 230
データベースサーバーオプション (dbeng12、dbsrv12), 230
FIPS プロトコルオプション
dbeng12 -ec, 223
dbsrv12 -ec, 223
説明, 385
fire_triggers オプション
SQL Anywhere SNMP Extension Agent OID, 1362
Transact-SQL 互換性オプション, 585
説明, 624
fire_triggers プロパティ
接続プロパティの説明, 713
first_day_of_week オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 625
first_day_of_week プロパティ
接続プロパティの説明, 713
FirstOption プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
FIXED ファイルフォーマット
input_format オプション, 890
Interactive SQL 出力, 898
for_xml_null_treatment オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 625
for_xml_null_treatment プロパティ
接続プロパティの説明, 713
force_view_creation オプション
説明, 626
force_view_creation オプション
SQL Anywhere SNMP Extension Agent OID, 1362
force_view_creation プロパティ
接続プロパティの説明, 713
ForceStart 接続パラメーター
説明, 345
FORCE 接続パラメーター
説明, 345
FreeBuffers 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
FreeBuffers プロパティ
サーバープロパティの説明, 754
FreePages プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
FullCompare 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
FullCompare プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
FunctionMaxParms プロパティ
サーバープロパティの説明, 754
FunctionMinParms プロパティ
サーバープロパティの説明, 754
FunctionName プロパティ
サーバープロパティの説明, 754

## G

GetData 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350

GetData プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
GetTypeInfoChar 接続パラメーター
ODBC 接続パラメーターの説明, 942
global_database_id オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 626
global_database_id プロパティ
接続プロパティの説明, 713
GlobalAutoincrement システムイベント
説明, 1128
GlobalDBID プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
go
Interactive SQL のデリミター, 887
使用法, 850
GRANT MEMBERSHIP IN GROUP 文
使用, 547
GRANT 文
DBA 権限, 542
DB 領域, 528
リソース権限, 542
WITH GRANT OPTION, 531
グループの作成, 546
グループメンバーシップ, 547
シーケンスジェネレーター, 528
テーブルパーミッション, 526
パスワードを持たない, 550
パーミッション, 526
プロシージャー, 527
GrowDB システムイベント
説明, 1128
GrowLog システムイベント
説明, 1128
例, 1129
GrowTemp システムイベント
説明, 1128
GSS-API ライブラリファイル
Kerberos, 63
Guest ユーザー
作成, 60

## H

HasCollationTailoring プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
HasEndianSwapFix プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
HashForcedPartitions プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
HashRowsFiltered プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
HashRowsPartitioned プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
HashWorkTables プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
HasNCHARLegacyCollationFix プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
HasTornWriteFix プロパティ
データベースプロパティの説明, 782
HeapsCarver 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
HeapsCarver プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
HeapsLocked 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
HeapsLocked プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
HeapsQuery 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
HeapsQuery プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713

データベースプロパティの説明, 782
HeapsRelocatable 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
HeapsRelocatable プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
Heimdal Kerberos クライアント
UNIX サポート, 63
Host 接続パラメーター
構文, 346
説明, 346
データベースミラーリング接続, 1163
トラブルシューティング, 84
host プロトコルオプション
説明, 381
HP-UX
IPv6 サポート, 100
SHLIB_PATH 環境変数, 453
HTML
Interactive SQL での表示, 871
HTML ファイルフォーマット
Interactive SQL 出力, 898
HTTP
クライアント API と文字セット変換, 480
サーバー要求ロギング, 388
サーバー設定, 294
サーバーログフォーマット, 389
プロトコルオプション, 370
http_connection_pool_basesize オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 627
http_connection_pool_basesize プロパティ
接続プロパティの説明, 713
http_connection_pool_timeout オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 628
http_connection_pool_timeout プロパティ
接続プロパティの説明, 713
http_session_timeout オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 629
http_session_timeout プロパティ
接続プロパティの説明, 713
HttpAddresses プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
HttpConnPoolCachedCount 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
HttpConnPoolCachedCount プロパティ
データベースプロパティの説明, 782
HttpConnPoolHits 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
HttpConnPoolHits プロパティ
データベースプロパティの説明, 782
HttpConnPoolMisses 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
HttpConnPoolMisses プロパティ
データベースプロパティの説明, 782
HttpConnPoolSteals 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
HttpConnPoolSteals プロパティ
データベースプロパティの説明, 782
HttpNumActiveReq プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
HttpNumConnections プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
HttpNumSessions プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
HttpPorts プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
HTTPS
Mobile Link トランスポートレイヤーセキュリティ, 1431
クライアント API と文字セット変換, 480
サーバー設定, 294
プロトコルオプション, 370
HttpsAddresses プロパティ

SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
HttpServiceName プロパティ
接続プロパティの説明, 713
HttpsNumActiveReq プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
HttpsNumConnections プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
HttpsPorts プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
HTTPS におけるトランスポートレイヤーセキュリティ
Mobile Link, 1431

## I

IANA
代替の文字セットエンコードラベル, 476
ポート番号, 396
IANA ラベル
文字セット, 507
iAnywhere.mib
説明, 1320, 1321
ロケーション, 1333
iAnywhere Solutions Oracle ドライバー
データソースの作成, 937
ICU
International Components for Unicode, 470
Unicode 照合アルゴリズム (UCA), 489
Windows Mobile での使用, 404
説明, 470
代替の文字セットエンコードラベル, 476
調整構文, 489
必要なときの判断, 470
文字セット変換で使用, 478
ICU ライブラリ
Windows Mobile での回避, 419
Windows Mobile でのデータベースのアンロード, 424
Windows Mobile 用データベースの作成, 419
identity_password プロトコルオプション
dbeng12 -ec, 223
dbsrv12 -ec, 223
説明, 384
IdentitySignature プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
identity プロトコルオプション
dbeng12 -ec, 223
dbsrv12 -ec, 223
説明, 383
IdleCheck 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
IdleCheck プロパティ
データベースプロパティの説明, 782
IdleChkpt 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
IdleChkpt プロパティ
データベースプロパティの説明, 782
IdleChkTime 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
IdleChkTime プロパティ
データベースプロパティの説明, 782
IdleTimeout プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
接続プロパティの説明, 713
IdleTime イベント
ポーリング, 1132
IdleWrite 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
IdleWrite プロパティ
データベースプロパティの説明, 782
Idle 接続パラメーター
説明, 348
ID ファイル
説明, 1421
IndAdd 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
IndAdd プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782

IndLookup 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
IndLookup プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
InitString 接続パラメーター
ODBC 接続パラメーターの説明, 942
INI ファイル
dbfhide を使用した単純暗号化の追加, 953
説明, 463
input_format オプション
Interactive SQL 設定, 884
説明, 890
INPUT 文
Interactive SQL での新しいローの挿入, 862
フォーマット, 890
INSERT パーミッション
説明, 520
付与, 526
INSERT 文
Interactive SQL での生成, 859
文字列のトランケーション, 691
INSTALL JAVA 文
Windows Mobile でサポート対象外, 436
installulnet.exe
Vista での権限の昇格の必要性, 188
Windows 7 での権限の昇格の必要性, 188
Windows Server 2008 での権限の昇格の必要性, 188
integrated_server_name オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 630
ドメインコントローラーサーバーで統合化ログインを指定する, 57
integrated_server_name プロパティ
接続プロパティの説明, 713
Integrated 接続パラメーター
説明, 349
Interactive SQL
.sql ファイルのデフォルトエディター, 847
dbisql 構文, 975
HTML, 871
Interactive SQL のオプションのアルファベット順リスト, 884
Mac OS X のハードウェア要件, 840
SQL 文, 878
SQL 文のアルファベット順リスト, 878
SQL 文のステップスルー, 848
Windows Mobile 上のデータベースの管理, 432
Windows Mobile 用データベースの作成, 421
XML, 871
イメージ, 869
印刷, 872
お気に入り, 854
オプション, 884
カスタマイズ, 846
起動, 841
行番号, 843
キーボードショートカット, 878
クエリエディターの表示, 878
グラフィカルなプラン, 867
警告メッセージの無効化, 847
計算カラムの更新, 862
結果セットのソート, 864
高速ランチャーの設定, 903
構文, 975
コマンドライン, 980
[コマンド履歴] ウィンドウ, 852
コメントの挿入, 850
スクリプトファイルの実行, 851
設定オプション, 884
説明, 840
ソフトウェア更新, 907
ソース制御の統合, 873
ソース制御プロジェクトを開く, 875
ソース制御を設定, 874
テキスト補完, 901
データの表示, 848
データベースへの接続, 841
データを表示した場合の予期しない記号, 479
テーブル値の編集, 859
テーブル編集の無効化, 861
テーブル、カラム、プロシージャーの検索, 857
認証アプリケーションで使用, 182
ファイルのチェックアウト, 876
ファイルのチェックイン, 877
ファイルへの読み書き用コードページの指定, 888
ファンクションキー, 878
複数のウィンドウを開く, 872
複数の文の実行, 850
文のキャンセル, 857
文の再呼び出し, 852

文の実行, 848
文の中断, 857
文の停止, 857
文のロギング, 856
ユーティリティ (dbisql ), 975
レポートされるエラー, 857
ローのコピー, 862
ローの削除, 862
ローの挿入, 861
Interactive SQL オプション
auto_commit, 885
auto_refetch, 886
bell, 886
command_delimiter, 887
commit_on_exit, 888
default_isql_encoding, 888
echo, 889
input_format, 890
isql_allow_read_client_file, 890
isql_allow_write_client_file, 891
isql_command_timing, 892
isql_escape_character, 893
isql_field_separator, 893
isql_maximum_displayed_rows, 895
isql_print_result_set, 895
isql_quote, 896
isql_show_multiple_result_sets, 896
nulls, 897
on_error, 898
output_format, 898
output_length, 899
output_nulls, 900
truncation_length, 900
初期設定, 574
設定, 884
分類, 575
Interactive SQL のオプション
Interactive SQL のオプションのアルファベット順リスト, 884
Interactive SQL ユーティリティ (dbisql)
構文, 975
サポートされるプラットフォーム, 980
終了コード, 981
interfaces ファイル
設定, 1438
Internet SCSI
データベースファイルの格納, 127
INT 接続パラメーター
説明, 349
IN キーワード
CREATE TABLE 文, 121
IOParallelism プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
IOToRecover 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
IOToRecover プロパティ
データベースプロパティの説明, 782
IPAddressMonitorPeriod プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
IPv4
説明, 99
IPv6
BCAST プロトコルオプションの使用, 101
IP プロトコルオプションの使用, 101
ME プロトコルオプションの使用, 101
MyIP プロトコルオプションの使用, 101
サポートされるプラットフォーム, 100
説明, 100
ブロードキャストプロトコルオプションの使用, 101
ホストプロトコルオプションの使用, 101
IPv6
インターフェイス識別子, 101
インターフェイス名, 101
IP アドレス
Open Server の設定, 1440
Windows Mobile デバイスでの特定, 415
IP プロトコルオプション
IPv6 アドレスの使用, 101
説明, 381
IQStore プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
iSCSI
データベースファイルの格納, 127
IsDebugger プロパティ
接続プロパティの説明, 713
IsEccAvailable プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341

サーバープロパティの説明, 754
IsFipsAvailable プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
IsIQ プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
IsNetworkServer プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
isolation_level オプション
SQL Anywhere SNMP Extension Agent OID, 1362
Sybase Open Client, 1443
ISOLATION_LEVEL 互換性オプション, 585
説明, 630
isolation_level プロパティ
接続プロパティの説明, 713
IsolationLevel 接続パラメーター
ODBC 接続パラメーターの説明, 942
IsPortableDevice プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
isql_allow_read_client_file オプション
Interactive SQL 設定, 884
説明, 890
isql_allow_write_client_file オプション
Interactive SQL 設定, 884
説明, 891
isql_command_timing オプション
Interactive SQL 設定, 884
説明, 892
isql_escape_character オプション
Interactive SQL 設定, 884
説明, 893
データのコピー, 862
isql_field_separator オプション
Interactive SQL 設定, 884
説明, 893
データのコピー, 862
isql_maximum_displayed_rows オプション
Interactive SQL 設定, 884
説明, 895
isql_print_result_set オプション
Interactive SQL 設定, 884
説明, 895
isql_quote オプション
Interactive SQL 設定, 884
説明, 896
データのコピー, 862
isql_show_multiple_result_sets オプション
Interactive SQL 設定, 884
説明, 896
IsRsaAvailable プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
IsRuntimeServer プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
IsService プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754

## J

Java
-cp サーバーオプション, 213
java_class_path オプション, 632
java_location オプション, 633
java_vm_options, 633
代替の文字セットエンコードラベル, 476
java_class_path オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 632
java_class_path プロパティ
接続プロパティの説明, 713
java_location オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 633
java_location プロパティ
接続プロパティの説明, 713
java_main_userid プロパティ
接続プロパティの説明, 713
java_vm_options オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 633

java_vm_options プロパティ
接続プロパティの説明, 713
JavaVM プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
java ディレクトリ
説明, 459
jConnect
Kerberos 認証, 67
TDS, 1435
Windows Mobile, 417
Windows Mobile で制限される機能, 435
アップグレードユーティリティ (dbupgrad), 1062
初期化ユーティリティ (dbinit), 959
jConnect メタデータサポート
Windows Mobile, 417
JDBC
ASE 互換性オプション, 585
クライアント API と文字セット変換, 480
データベース互換性オプション, 585

## K

keep-alive request-header フィールド
KeepaliveTimeout 値の設定, 386
KeepaliveTimeout プロトコルオプション
説明, 386
Kerberos
CSFC5KTNAME 環境変数, 63
GSS-API ライブラリファイル, 63
jConnect 接続, 67
Kerberos 接続パラメーター (KRB), 350
Kerberos プリンシパル, 64
keytab ファイル, 63
-kl オプション, 250
KRB5_KTNAME 環境変数, 63
-krb オプション, 253
-krp オプション, 251
-kr オプション, 252
login_mode オプション, 635
Sybase Open Client 接続, 67
Windows Mobile でサポート対象外, 435
Windows での SSPI の使用, 68
キー配布センター (KDC), 64
クライアント, 63
セキュリティについての考慮事項, 60
接続のトラブルシューティング, 69
説明, 61
チケット付与権限付きチケット, 65
テンポラリオプション, 73
パーミッションの取り消し, 68
パーミッションの付与, 67
Kerberos キー配布センター (KDC)
説明, 64
Kerberos 接続パラメーター
説明, 350
Kerberos 認証
SQL Anywhere クライアントでサポートされている SSPI, 62
SQL Anywhere サーバーでサポートされていない SSPI, 62
Kerberos プリンシパル
説明, 64
Kerberos ログインパーミッションの取り消し
説明, 68
Kerberos ログインマッピングの作成
説明, 67
KeysInSQLStatistics 接続パラメーター
ODBC 接続パラメーターの説明, 942
keytab ファイル
デフォルトロケーション, 63
KRB5_KTNAME 環境変数
Kerberos, 63
KRB 接続パラメーター
説明, 350
KTO プロトコルオプション
説明, 386

## L

Language 接続パラメーター
説明, 351
Language プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341, 1355
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
LANG 接続パラメーター
説明, 351
LastCheckpointTime プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782

LastCommitRedoPos プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
接続プロパティの説明, 713
データベースプロパティの説明, 782
LastConnectionProperty プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
LastDatabaseProperty プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
LastIdle プロパティ
接続プロパティの説明, 713
LastOption プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
LastPlanText プロパティ
接続プロパティの説明, 713
LastReqTime プロパティ
接続プロパティの説明, 713
LastServerProperty プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
LastStatement プロパティ
接続プロパティの説明, 713
LastSyncedRedoPos プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
LastWrittenRedoPos プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
LazyAutocommit 接続パラメーター
ODBC 接続パラメーターの説明, 942
LazyClose 接続パラメーター
説明, 352
LCLOSE 接続パラメーター
説明, 352
LD_LIBRARY_PATH 環境変数
説明, 445
LDAP
AIX, 103
LDAP プロトコルオプション, 387
saldap.ini, 104
Windows Mobile でサポート対象外, 435
サーバー列挙ユーティリティ (dblocate), 1002
接続, 103
パーソナルサーバーとネットワークサーバーの違い, 142
LDAP プロトコルオプション
説明, 387
LegalCopyright プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
LegalTrademarks プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
LF プロトコルオプション
説明, 389
libctl.cfg
DSEdit, 1439
LIBPATH 環境変数
説明, 446
LicenseCount プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
LicensedCompany プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
LicensedUser プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
LicenseKey プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
LicenseType プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
LINKS 接続パラメーター
IPv6 アドレス, 100
オプション, 97
カッコ, 478
説明, 327
Linux

dbconsole の起動, 905
Interactive SQL の起動, 841
IPv6 アドレスに必要なインターフェイス識別子, 101
IPv6 サポート, 100
LD_LIBRARY_PATH 環境変数, 445
SELinux のポリシー, 1380
Sybase Central の起動, 819
サンプルデータベースへの接続, 7
[サーバー起動オプション] ウィンドウのサーバーメッセージの表示, 284
[サーバー起動オプション] ウィンドウの使用, 286
シェルモードでのサーバーメッセージの表示, 282
スレッドの動作, 163
データベースサーバーメッセージウィンドウの表示, 286
非同期 I/O の使用の無効化, 281
Linux サービス
データベースサーバー, 170
LivenessTimeout 接続パラメーター
説明, 352
LivenessTimeout プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
接続プロパティの説明, 713
LOAD TABLE 文
セキュリティ, 1396
データベースミラーリングでの使用の制限, 1140
localhost
Host (IP) プロトコルオプション, 381
Host 接続パラメーター, 346
Open Server の設定, 1440
指定, 412
プロキシポート, 413
LocalOnly プロトコルオプション
説明, 387
LocalSystem アカウント
オプション, 175
説明, 171
Location レジストリエントリ
Windows でのファイル検索, 461
lock_rejected_rows オプション
SQL Anywhere SNMP Extension Agent OID, 1362
lock_rejected_rows プロパティ
接続プロパティの説明, 713
LockCount 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
LockCount プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
LockedCursorPages 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
LockedCursorPages プロパティ
サーバープロパティの説明, 754
LockedHeapPages 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
LockedHeapPages プロパティ
サーバープロパティの説明, 754
LockIndexID プロパティ
接続プロパティの説明, 713
LockName プロパティ
接続プロパティの説明, 713
LockRowID プロパティ
接続プロパティの説明, 713
LockTableOID プロパティ
接続プロパティの説明, 713
LockTablePages 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
LockTablePages プロパティ
データベースプロパティの説明, 782
LOCK TABLE 文
blocking_others_timeout オプション, 602
log_deadlocks オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 634
log_deadlocks プロパティ
接続プロパティの説明, 713
LogDiskSpace システムイベント
説明, 1127
LogFileFragments プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
LogFile 接続パラメーター
説明, 354
LogFile プロトコルオプション

説明, 388
LogFormat プロトコルオプション
説明, 389
LogFreeCommit 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
LogFreeCommit プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
login_mode
Integrated (INT) 接続パラメーター, 349
login_mode オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 635
統合化ログイン, 52
login_mode プロパティ
接続プロパティの説明, 713
login_procedure オプション
RAISERROR による接続の不許可, 637
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 636
パスワード有効期限の導入, 1383
login_procedure プロパティ
接続プロパティの説明, 713
LoginTime プロパティ
接続プロパティの説明, 713
LogMaxSize プロトコルオプション
説明, 390
LogMirrorName プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
LogName プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
LogOptions プロトコルオプション
説明, 391
LogWrite 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
LogWrite プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
LOG 接続パラメーター
説明, 354
LOG プロトコルオプション
説明, 388
LOPT プロトコルオプション
説明, 391
LO プロトコルオプション
説明, 387
LSIZE プロトコルオプション
説明, 390
LTMGeneration プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
LTMTrunc プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
LTO 接続パラメーター
説明, 352

## M

MachineName プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
Mac OS X
dbconsole の起動, 906
dbconsole ユーティリティのハードウェア要件, 906
DYLD_LIBRARY_PATH 環境変数, 445
Interactive SQL の起動, 843
Interactive SQL のハードウェア要件, 840
IPv6 サポート, 100
ODBC データソースの作成, 45
Sybase Central の起動, 818
Sybase Central のハードウェア要件, 817
環境変数の設定, 443
サンプル, 444
サンプルデータベースへの接続, 9
サンプルの設定, 444
サーバーメッセージの表示, 284
推奨照合, 506
ファイルのソース指定, 444
MAGIC
user_estimates オプション, 706
MainHeapBytes 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337

MainHeapBytes プロパティ
サーバープロパティの説明, 754
MainHeapPages 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
MainHeapPages プロパティ
サーバープロパティの説明, 754
Management Information Base
説明, 1321
MapPhysicalMemoryEng 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
MapPhysicalMemoryEng プロパティ
サーバープロパティの説明, 754
materialized_view_optimization オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 639
materialized_view_optimization プロパティ
接続プロパティの説明, 713
MatView 接続パラメーター
説明, 356
max_client_statements_cached オプション
SQL Anywhere SNMP Extension Agent OID, 1362
クライアントでの文のキャッシュで使用, 642
説明, 640
max_client_statements_cached プロパティ
接続プロパティの説明, 713
max_cursor_count オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 642
max_cursor_count プロパティ
接続プロパティの説明, 713
max_failed_login_attempts オプション
DBA ログイン動作, 538
max_hash_size オプション
SQL Anywhere SNMP Extension Agent OID, 1362
max_hash_size プロパティ
接続プロパティの説明, 713
max_plans_cached オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 643
max_plans_cached プロパティ
接続プロパティの説明, 713
max_priority オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 643
max_priority プロパティ
接続プロパティの説明, 713
max_query_tasks オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 644
max_query_tasks プロパティ
接続プロパティの説明, 713
max_recursive_iterations オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 645
max_recursive_iterations プロパティ
接続プロパティの説明, 713
max_retry_connect_time オプション
使用, 1200
max_statement_count オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 646
max_statement_count プロパティ
接続プロパティの説明, 713
max_temp_space オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 647
max_temp_space プロパティ
接続プロパティの説明, 713
MaxCacheSize 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
MaxCacheSize プロパティ
サーバープロパティの説明, 754
MaxConnections プロトコルオプション
説明, 392
MaxConnections プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
MAXCONN プロトコルオプション
説明, 392
MaxEventType プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341

サーバープロパティの説明, 754
MaxIO 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
MaxIO プロパティ
データベースプロパティの説明, 782
MaxMessage プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
MaxMultiProgrammingLevel プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
MaxRead 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
MaxRead プロパティ
データベースプロパティの説明, 782
MaxRemoteCapability プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
MaxRequestSize プロトコルオプション
説明, 393
MAXSIZE プロトコルオプション
説明, 393
MaxWrite 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
MaxWrite プロパティ
データベースプロパティの説明, 782
Message Agent (dbremote)
トランザクションログの管理, 1107
MessageCategoryLimit プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
MessageReceived プロパティ
接続プロパティの説明, 713
MessageText プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
MessageTime プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
MessageWindowSize プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
Message プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
MESSAGE 文
debug_messages オプションの設定, 616
ME プロトコルオプション
IPv6 アドレスの使用, 101
説明, 394
MIB
SQL Anywhere SNMP Extension Agent でサポートされる MIB, 1321
定義, 1321
Microsoft Access
TIMESTAMP の比較, 619
MIME
代替の文字セットエンコードラベル, 476
min_password_length オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 648
パスワードセキュリティの強化, 1383
min_password_length プロパティ
接続プロパティの説明, 713
MinCacheSize 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
MinCacheSize プロパティ
サーバープロパティの説明, 754
MinMultiProgrammingLevel プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
MirrorFailover システムイベント
使用, 1173
説明, 1128
MirrorMode プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
データベースミラーリングの同期実行モードの決定, 1143
MirrorRole プロパティ


 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
MirrorServerDisconnect システムイベント
使用, 1173
説明, 1128
MirrorServerName パラメーター
使用, 1173
MirrorServerState プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
MirrorServerWaits 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
MirrorServerWaits プロパティ
データベースプロパティの説明, 782
MirrorState プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
MIT Kerberos クライアント
UNIX サポート, 63
Windows サポート, 63
mlasinst ユーティリティ
Vista での権限の昇格の必要性, 188
Windows 7 での権限の昇格の必要性, 188
Windows Server 2008 での権限の昇格の必要性, 188
mlsrv12
Linux サービスとして実行, 1007
Windows サービスとして実行, 1013
サーバートランスポートレイヤーセキュリティを使用する起動, 1427
ライセンス, 1003
Mobile Link
高可用性, 1170
データベースオプション, 587
Mobile Link クライアント
データベースオプション, 587
Mobile Link クライアント／サーバー通信の暗号化
説明, 1426
Mobile Link サーバーの確認
Mobile Link トランスポートレイヤーセキュリティ, 1429
Mobile Link サーバーの起動
トランスポートレイヤーセキュリティ, 1427
Mobile Link サーバーのモニタリング
Mobile Link、説明, 1217
Mobile Link サーバーファーム
Mobile Link モニタリング, 1217
Mobile Link 同期
default_timestamp_increment の設定, 619
truncate_timestamp_values の設定, 702
バックアップ, 1107
Mobile Link トランスポートレイヤーセキュリティ
説明, 1410
Mobile Link ユーティリティ
証明書作成 (createcert) の構文, 924
証明書ビューアー (viewcert) の構文, 934
MultiByteCharSet プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
MultiPacketsReceived 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
MultiPacketsReceived プロパティ
サーバープロパティの説明, 754
MultiPacketsSent 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
MultiPacketsSent プロパティ
サーバープロパティの説明, 754
MultiPageAllocs 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
MultiPageAllocs プロパティ
サーバープロパティの説明, 754
MultiProgrammingLevel プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
MyIP プロトコルオプション
IPv6 アドレスの使用, 101
説明, 394

## N

Name プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341, 1355
サーバープロパティの説明, 754
接続プロパティの説明, 713

データベースプロパティの説明, 782
NAS
データベースファイルの格納, 127
NativeProcessorArchitecture プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
nchar_charset
説明, 476
nchar_charset エイリアス
説明, 477
NcharCharSet プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
接続プロパティの説明, 713
データベースプロパティの説明, 782
NcharCollation プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
NCHAR 照合
説明, 491
NCHAR データ型
照合順, 959
NDS
ファイル名, 155
nearest_century オプション
SQL Anywhere SNMP Extension Agent OID, 1362
Transact-SQL 互換性オプション, 585
説明, 649
nearest_century プロパティ
接続プロパティの説明, 713
net.cfg
クライアント／サーバー通信のトラブルシューティング, 95
NewPassword 接続パラメーター
説明, 355
パスワード有効期限の導入, 1383
NEWPWD 接続パラメーター
説明, 355
パスワード有効期限の導入, 1383
NextScheduleTime プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
NIST
FIPS 証明書, 1412
NodeAddress プロパティ
接続プロパティの説明, 713
NodeType 接続パラメーター
説明, 357
NODE 接続パラメーター
説明, 357
non_keywords オプション
SQL Anywhere SNMP Extension Agent OID, 1362
Transact-SQL 互換性オプション, 585
説明, 650
non_keywords プロパティ
接続プロパティの説明, 713
NULL
ANSI の動作, 597
nulls オプション, 897
Transact-SQL 動作, 597
エクスポート用の定義, 900
nulls オプション
Interactive SQL 設定, 884
説明, 897
NULL 値
カラムでの許可, 115
Number プロパティ
接続 ID の取得, 30
接続プロパティの説明, 713
NumLogicalProcessorsUsed プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
NumLogicalProcessors プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
NumPhysicalProcessorsUsed プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
NumPhysicalProcessors プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754

## O

ObjectType プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341

サーバープロパティの説明, 754
ODBC
dbdsn ユーティリティを使用した作成, 42
Delphi, 650
odbc_describe_binary_as_varbinary オプション, 650
odbc_distinguish_char_and_varchar オプション, 651
ODBC データソースアドミニストレーターを使用した作成, 39
UNIX サポート, 48
UNIX 用の初期化ファイル, 48
Windows Mobile で制限される機能, 435
Windows Mobile でのデータソースの使用, 47
アドミニストレーター, 39
クライアント API と文字セット変換, 480
接続, 49
[接続] ウィンドウを使用した作成, 38
接続パラメーター, 320
データソース, 36
データソース接続パラメーター, 942
ドライバーのロケーション, 77
トラブルシューティング, 994
odbc_describe_binary_as_varbinary オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 650
odbc_describe_binary_as_varbinary プロパティ
接続プロパティの説明, 713
odbc_distinguish_char_and_varchar オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 651
odbc_distinguish_char_and_varchar プロパティ
接続プロパティの説明, 713
ODBC_INI 環境変数
システム情報ファイルの検索, 48
説明, 447
odbc.ini
説明, 48
ODBCHOME 環境変数
説明, 447
ODBCINI 環境変数
システム情報ファイルの検索, 48
説明, 447
ODBC INI ファイル
説明, 48
ODBC Sample
使用, 410
ODBC 接続パラメーター
Delphi, 942
DescribeCursor, 942
Driver, 942
GetTypeInfoChar, 942
InitString, 942
IsolationLevel, 942
KeysInSQLStatistics, 942
LazyAutocommit, 942
PrefetchOnOpen, 942
PreventNotCapable, 942
SuppressWarnings, 942
TranslationDLL, 942
TranslationName, 942
TranslationOption, 942
説明, 942
ODBC データソース
dbdsn を使用して作成, 936
Mac OS X での作成, 45
ODBC データソースアドミニストレーターを使用した作成, 39
[ODBC データソースとして保存] を使用した作成, 38
UNIX, 48
Windows Mobile 用に作成, 412
生成, 38
設定, 39
説明, 36
ODBC データソースアドミニストレーター
使用, 39
ODBC データソースとして保存
説明, 38
[ODBC データソースと接続]
例, 18
[ODBC データソースを使用した接続]
[サンプルデータベース] ウィンドウ, 30
[接続] ウィンドウ, 14
ODBC ドライバー
設定, 45
oem_string オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 652
oem_string プロパティ
接続プロパティの説明, 713
OEM Edition
説明, 181

OEM コードページ
説明, 476
OID
RDBMS MIB, 1372
SQL Anywhere MIB, 1333
サーバー統計, 1337
サーバープロパティ, 1341
説明, 1321
定義, 1321
データベースオプション, 1362
データベース統計, 1350
データベースプロパティ, 1355
OLAP
optimization_workload オプション, 658
OLE DB
クライアント API と文字セット変換, 480
OmniConnect サポート
説明, 1436
OmniIdentifier プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
on_charset_conversion_failure オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 654
on_charset_conversion_failure プロパティ
接続プロパティの説明, 713
on_error オプション
説明, 898
on_tsql_error オプション
ASE 互換性オプション, 585
SQL Anywhere SNMP Extension Agent OID, 1362
Sybase Open Client, 1443
Transact-SQL 互換性オプション, 585
説明, 655
on_tsql_error プロパティ
接続プロパティの説明, 713
ON EXCEPTION RESUME 句
on_tsql_error オプション, 655
OPENSTRING 句
ファイルの問い合わせに必要なパーミッション, 540
optimization_goal オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 655
optimization_goal プロパティ
接続プロパティの説明, 713
optimization_level オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 657
optimization_level プロパティ
接続プロパティの説明, 713
optimization_workload オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 658
optimization_workload プロパティ
接続プロパティの説明, 713
OptionWatchAction プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
説明, 573
データベースプロパティの説明, 782
OptionWatchList プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
説明, 573
データベースプロパティの説明, 782
Oracle ドライバー
データソースの作成, 937
os_charset エイリアス
説明, 477
OSUser プロパティ
接続プロパティの説明, 713
Outbound Enabler
Linux サービスとして実行, 1007
Windows サービスとして実行, 1013
高可用性とスケールアウトソリューション, 1205
output_format オプション
Interactive SQL 設定, 884
説明, 898
output_length オプション
Interactive SQL 設定, 884
説明, 899
output_nulls オプション
Interactive SQL 設定, 884
説明, 900
Override-Magic
user_estimates オプション, 706

## P


PacketSize プロパティ
接続プロパティの説明, 713
PacketsReceivedUncomp 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
PacketsReceivedUncomp プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
PacketsReceived 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
PacketsReceived プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
PacketsSentUncomp 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
PacketsSentUncomp プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
PacketsSent 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
PacketsSent プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
PageRelocations 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
PageRelocations プロパティ
データベースプロパティの説明, 782
PageSize プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341, 1355
サーバープロパティの説明, 754
データベースプロパティの説明, 782
page モード
データベースミラーリング, 1143
ParentConnection プロパティ
接続プロパティの説明, 713
PartnerState プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
Password 接続パラメーター
説明, 360
password パラメーター
LDAP, 105
PATH 環境変数
説明, 447
PBUF 接続パラメーター
説明, 361
PeakCacheSize 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
PeakCacheSize プロパティ
サーバープロパティの説明, 754
PEM コード化 PKI オブジェクト
表示, 934
Perl
クライアント API と文字セット変換, 480
PHP
クライアント API と文字セット変換, 480
PID ファイル
Linux サービス, 1011
pinging
サーバー, 994
ping ユーティリティ (dbping)
Sybase Open Client のテスト, 1442
構文, 994
終了コード, 997
使用, 93
pinned_cursor_percent_of_cache オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 659
pinned_cursor_percent_of_cache プロパティ
接続プロパティの説明, 713
PKI オブジェクト
表示, 934
PlatformVer プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
Platform プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
port パラメーター

LDAP, 105
PORT プロトコルオプション
説明, 396
post_login_procedure オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 659
パスワード有効期限の導入, 1383
post_login_procedure プロパティ
接続プロパティの説明, 713
precision オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 661
precision プロパティ
接続プロパティの説明, 713
PrefetchBuffer 接続パラメーター
説明, 361
PrefetchOnOpen 接続パラメーター
ODBC 接続パラメーターの説明, 942
説明, 362
PrefetchRows 接続パラメーター
説明, 363
prefetch オプション
DisableMultiRowFetch 接続パラメーター, 339
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 662
prefetch プロパティ
接続プロパティの説明, 713
Prepares 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
Prepares プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
PrepStmt 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
PrepStmt プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
preserve_source_format オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 663
preserve_source_format プロパティ
接続プロパティの説明, 713
prevent_article_pkey_update オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 664
prevent_article_pkey_update プロパティ
接続プロパティの説明, 713
PreventNotCapable 接続パラメーター
ODBC 接続パラメーターの説明, 942
priority オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 664
priority プロパティ
接続プロパティの説明, 713
ProcedurePages 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
ProcedurePages プロパティ
データベースプロパティの説明, 782
ProcedureProfiling プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
ProcessCPUSystem プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
ProcessCPUUser プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
ProcessCPU プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
ProcessID プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
ProcessorArchitecture プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
ProductName プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754

ProductVersion プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
ProfileFilterConn プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
ProfileFilterUser プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
progress_messages オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 665
progress_messages プロパティ
接続プロパティの説明, 713
Progress プロパティ
接続プロパティの説明, 713
PROPERTY 関数
データベースサーバープロパティのアルファベット順リスト, 753
PROWS 接続パラメーター
説明, 363
PUBLIC オプション
DBA 権限が必要, 570
説明, 570
PUBLIC グループ
説明, 551
PWD 接続パラメーター
説明, 360
Python
クライアント API と文字セット変換, 480

## Q

qualify_owners オプション
SQL Remote 説明, 667
SQL Remote レプリケーションオプション, 588
query_mem_timeout オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 667
query_mem_timeout プロパティ
接続プロパティの説明, 713
query_plan_on_open オプション
Transact-SQL 互換性オプション, 585
QueryBypassedCosted プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
QueryBypassedHeuristic プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
QueryBypassedOptimized プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
QueryBypassed プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
QueryCachedPlans プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
QueryCachePages 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
QueryCachePages プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
QueryDescribedBypass プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
QueryDescribedOptimizer プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
QueryHeapPages 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
QueryHeapPages プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
QueryJHToJNLOptUsed プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
QueryLowMemoryStrategy 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
QueryLowMemoryStrategy プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
QueryMemActiveCurr 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
QueryMemActiveCurr プロパティ
サーバープロパティの説明, 754

接続プロパティの説明, 713
データベースプロパティの説明, 782
QueryMemActiveEst 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
QueryMemActiveEst プロパティ
サーバープロパティの説明, 754
QueryMemActiveMax プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
QueryMemExtraAvail 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
QueryMemExtraAvail プロパティ
サーバープロパティの説明, 754
QueryMemGrantBaseMI プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
QueryMemGrantBase プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
QueryMemGrantExtra プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
QueryMemGrantFailed 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
QueryMemGrantFailed プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
QueryMemGrantGranted 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
QueryMemGrantGranted プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
QueryMemGrantRequested 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
QueryMemGrantRequested プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
QueryMemGrantWaited 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
QueryMemGrantWaited プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
QueryMemGrantWaiting 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
QueryMemGrantWaiting プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
QueryMemPages プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
QueryMemPercentOfCache プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
QueryOpened プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
QueryOptimized プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
QueryReused プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
QueryRowsFetched プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
QueryRowsMaterialized 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
QueryRowsMaterialized プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
QuittingTime プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
quote_all_identifiers オプション
SQL Remote 説明, 668
SQL Remote レプリケーションオプション, 588

quoted_identifier オプション
SQL Anywhere SNMP Extension Agent OID, 1362
Sybase Open Client, 1443
Transact-SQL 互換性オプション, 585
説明, 668
quoted_identifier プロパティ
接続プロパティの説明, 713

## R

RAISERROR システムイベント
説明, 1128
RAISERROR 文
continue_after_raiserror オプション, 609
on_tsql_error オプション, 655
RAS
ダイアルアップネットワーキング, 102
RCVBUFSZ プロトコルオプション
説明, 395
rdbmsDbInfoTable
説明, 1373, 1375
rdbmsDbLimitedResourceTable
説明, 1374
rdbmsDbParamTable
説明, 1374
rdbmsDbTable
説明, 1373
RDBMS MIB
説明, 1324
テーブルのリスト, 1372
RDBMS-MIB.mib
SQL Anywhere SNMP Extension Agent, 1320
説明, 1324
ロケーション, 1372
rdbmsSrvInfoTable
説明, 1376
rdbmsSrvLimitedResourceTable
説明, 1378
rdbmsSrvParamTable
説明, 1377
read_authdn パラメーター
LDAP, 105
read_password パラメーター
LDAP, 105
read_past_deleted オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 669
read_past_deleted プロパティ
接続プロパティの説明, 713
READCLIENTFILE 権限
継承可能, 540
説明, 540
付与, 542
READFILE 権限
継承可能, 540
説明, 540
付与, 542
ReadOnly プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
ReceiveBufferSize プロトコルオプション
説明, 395
ReceivingTracingFrom プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
recovery_time オプション
SQL Anywhere SNMP Extension Agent OID, 1362
使用, 1114
説明, 669
recovery_time プロパティ
接続プロパティの説明, 713
RecoveryUrgency 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
RecoveryUrgency プロパティ
データベースプロパティの説明, 782
RecursiveIterationsHash プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
RecursiveIterationsNested プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
RecursiveIterations プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
RecursiveJNLMisses プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
RecursiveJNLProbes プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782

REFERENCES パーミッション
説明, 520
付与, 526
Relay Server
Relay Server、モニタリング, 1217
高可用性とスケールアウトソリューション, 1205
Relay Server ステイトマネージャー
Linux サービスとして実行, 1007
Relay Server ステイトマネージャー
Windows サービスとして実行, 1013
Relay Server ファーム
Mobile Link、モニタリング, 1217
RelocatableHeapPages 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
RelocatableHeapPages プロパティ
データベースプロパティの説明, 782
RememberLastPlan プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
RememberLastStatement プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
remote_idle_timeout オプション
SQL Anywhere SNMP Extension Agent OID, 1362
remote_idle_timeout プロパティ
接続プロパティの説明, 713
RemoteCapability プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
REMOTE DBA 権限
オンラインバックアップに必要, 1071
説明, 540
RemoteputWait 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
RemoteputWait プロパティ
サーバープロパティの説明, 754
RemoteTrunc プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
REMOTE パーミッション
SQL Remote の付与と取り消し, 535
REMOVE JAVA 文
Windows Mobile でサポート対象外, 436
REORGANIZE TABLE 文
Windows Mobile でサポート対象外, 436
replicate_all オプション
SQL Anywhere SNMP Extension Agent OID, 1362
replicate_all プロパティ
接続プロパティの説明, 713
replication_error_piece オプション
SQL Remote 説明, 671
SQL Remote レプリケーションオプション, 588
replication_error オプション
SQL Remote 説明, 671
SQL Remote レプリケーションオプション, 588
ReqCountActive プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341, 1355
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
ReqCountBlockContention プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341, 1355
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
ReqCountBlockIO プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341, 1355
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
ReqCountBlockLock プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341, 1355
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
ReqCountUnscheduled プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341, 1355
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
ReqStatus プロパティ
接続プロパティの説明, 713

ReqTimeActive プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341, 1355
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
ReqTimeBlockContention プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341, 1355
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
ReqTimeBlockIO プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341, 1355
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
ReqTimeBlockLock プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341, 1355
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
ReqTimeUnscheduled プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341, 1355
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
ReqType プロパティ
接続プロパティの説明, 713
request_timeout オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 672
request_timeout プロパティ
接続プロパティの説明, 713
RequestFilterConn プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
RequestFilterDB プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
RequestLogFile プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
RequestLogging プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
RequestLogMaxSize プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
RequestLogNumFiles プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
RequestsReceived 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
RequestsReceived プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
RequestTiming プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
Req 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
Req プロパティ
サーバープロパティの説明, 754
reserved_keywords オプション
SQL Anywhere SNMP Extension Agent OID, 1362
Transact-SQL 互換性オプション, 585
説明, 673
reserved_keywords プロパティ
接続プロパティの説明, 713
リソース権限
継承不可能, 541
説明, 541
付与, 542
RetryConnectionTimeout 接続パラメーター
説明, 364
ミラーリングされたデータベースへの接続, 1163
RetryConnTO 接続パラメーター
説明, 364

ミラーリングされたデータベースへの接続, 1163
return_date_time_as_string オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 674
return_date_time_as_string プロパティ
接続プロパティの説明, 713
Rlbk 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337, 1350
Rlbk プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
rollback_on_deadlock オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 674
rollback_on_deadlock プロパティ
接続プロパティの説明, 713
RollbackLogPages 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
RollbackLogPages プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
ROLLBACK 文
Interactive SQL のショートカット, 851
WITH HOLD カーソル, 593
ログ, 1116
row_counts オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 675
row_counts プロパティ
接続プロパティの説明, 713
RSA
サポート, 1411
RSA オプション
dbeng12 -ec, 223
dbsrv12 -ec, 223
RSA 証明書
作成, 924
表示, 934
Ruby
クライアント API と文字セット変換, 480

## S

sa_config.csh
ソース指定, 444
sa_config.sh
ソース指定, 444
sa_conn_properties システムプロシージャー
使用, 571
接続プロパティのアルファベット順リスト, 713
sa_db_properties システムプロシージャー
データベースプロパティのアルファベット順リスト, 781
SA_DEBUG グループ
説明, 551
sa_eng_properties システムプロシージャー
データベースサーバープロパティのアルファベット順リスト, 753
sa_monitor_connection_failed_event イベントモニター, 1311
sa_monitor_connection_failure テーブルモニター, 1311
sa_monitor_count_unsubmitted_crash_reports 関数モニター, 1311
sa_monitor_user
説明, 1311
sa_server_option システムプロシージャー
データベースオプション設定のモニタリング, 573
SACA
UTF-8 文字セット, 488
シングルバイト文字セット, 488
シングルバイト文字セットでの使用, 489
説明, 488
マルチバイト文字セット, 488
SACHARSET 環境変数
文字セットの指定, 449
SADatabase エージェント
設定, 1186
テスト, 1187
saDbOptMetaDataTable
SQL Anywhere MIB, 1337
saDbPropMetaDataTable
SQL Anywhere MIB, 1336
saDbStatMetaDataTable
SQL Anywhere MIB, 1336
SADIAGDIR 環境変数
診断情報のロケーションの指定, 449

SALANG 環境変数
言語の指定, 450
saldap.ini
説明, 104
SALOGDIR 環境変数
説明, 450
samonitor.bat
Mobile Link によるモニターサービスの開始, 1236
Relay Server によるモニターサービスの開始, 1236
モニターサービスの開始, 1236
モニターサービスの停止, 1239
samonitor.sh
Mobile Link によるモニターサービスの開始, 1236
Relay Server によるモニターサービスの開始, 1236
モニターサービスの開始, 1236
sample_config.csh
ソース指定, 444
sample_config.sh
ソース指定, 444
SAN
データベースファイルの格納, 127
SAPLAN ファイル
プランビューアーのファイル拡張子, 867
SAServer エージェント
設定, 1184
テスト, 1185
sasnmp.ini
SQL Anywhere SNMP Extension Agent に必要, 1325
説明, 1320
sasrv.ini
CommLinks 接続パラメーター, 87
Host 接続パラメーター, 84
サーバー起動のトラブルシューティング, 92
サーバー情報, 91
saSrvPropMetaDataTable
SQL Anywhere MIB, 1335
saSrvStatMetaDataTable
SQL Anywhere MIB, 1334
SATMP 環境変数
UNIX, 457
UNIX で保護された共有メモリ接続, 1381
説明, 451
ディスク領域, 1211
save_remote_passwords オプション
SQL Remote のオプション, 676
scale オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 676
scale プロパティ
接続プロパティの説明, 713
scjview
説明, 816
scjview.exe
高速ランチャーオプション, 903
説明, 816
scripts ディレクトリ
説明, 459
search_timeout パラメーター
LDAP, 105
secure_feature_key オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 677
secure_feature_key プロパティ
接続プロパティの説明, 713
SecureFeatures プロパティ
-sf サーバーオプション, 271
SELECT パーミッション
説明, 520
付与, 526
SELinux のポリシー
SQL Anywhere のポリシーの使用, 1380
SendBufferSize プロトコルオプション
説明, 395
SendFail 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
SendFail プロパティ
サーバープロパティの説明, 754
SendingTracingTo プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
ServerEdition プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
ServerIdle システムイベント
説明, 1128
例, 1130

ServerName 接続パラメーター
組み込みデータベース, 28
説明, 365
ミラーリングされたデータベースへの接続, 1163
文字セット, 34
ServerName プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
ServerNodeAddress プロパティ
接続プロパティの説明, 713
ServerPort プロトコルオプション
説明, 396
ServerPort プロパティ
接続プロパティの説明, 713
Server 接続パラメーター
説明, 365
ミラーリングされたデータベースへの接続, 1163
Server 接続パラメーター
組み込みデータベース, 28
文字セット, 34
server パラメーター
LDAP, 105
SessionCreateTime プロパティ
接続プロパティの説明, 713
SessionID プロパティ
接続プロパティの説明, 713
SessionLastTime プロパティ
接続プロパティの説明, 713
SessionTimeout プロパティ
接続プロパティの説明, 713
SET OPTION 文
Interactive SQL オプション, 884
使用, 568
SET TEMPORARY OPTION 文
使用, 568
SetupVSPackage.exe
Vista での権限の昇格の必要性, 188
Windows 7 での権限の昇格の必要性, 188
Windows Server 2008 での権限の昇格の必要性, 188
SHLIB_PATH 環境変数
説明, 453
SMP
-gt dbeng12/dbsrv12 サーバーオプション, 244
プロセッサーの数, 158
SnapshotCount 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
SnapshotCount プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
SnapshotIsolationState プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
SNDBUFSZ プロトコルオプション
説明, 395
SNMP
SQL Anywhere SNMP Extension Agent の使用, 1319
インストール, 1325
エージェント, 1320
説明, 1320
動的トラップ, 1331
トラップ, 1320
トラップの使用, 1330
マネージャー, 1320
SNMPv2-SMI.mib
説明, 1320
SNMPv2-TC.mib
説明, 1320
SNMP サービス
再起動, 1325
Solaris
IPv6 サポート, 100
LD_LIBRARY_PATH 環境変数, 445
サンプルデータベースへの接続, 7
シェルモードでのサーバーメッセージの表示, 282
sort_collation オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 678
sort_collation プロパティ
接続プロパティの説明, 713
SortMergePasses プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
SortRowsMaterialized プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
SortRunsWritten プロパティ
接続プロパティの説明, 713

データベースプロパティの説明, 782
SortSortedRuns プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
SortWorkTables プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
sp_login_environment システムプロシージャー
login_procedure オプションでの使用, 636
SQL
Windows Mobile でサポート対象外の文, 436
sql_flagger_error_level オプション
SQL Anywhere SNMP Extension Agent OID, 1362
Transact-SQL 互換性オプション, 585
説明, 679
sql_flagger_error_level プロパティ
接続プロパティの説明, 713
sql_flagger_warning_level オプション
SQL Anywhere SNMP Extension Agent OID, 1362
Transact-SQL 互換性オプション, 585
説明, 680
sql_flagger_warning_level プロパティ
接続プロパティの説明, 713
SQL/2008 準拠
SQL_FLAGGER_ERROR オプション, 679
更新, 596
sql.ini
設定, 1438
SQLANY12 環境変数
説明, 453
SQLANYSAMP12 環境変数
説明, 454
SQL Anywhere
Open Server として設定, 1436
SQL Anywhere のインストール, 404
SQL Anywhere の使用, 403
Vista での実行, 187
Windows 7 での実行, 187
Windows Mobile でサポート対象外の機能, 435
Windows Mobile での使用, 403
Windows Mobile のデータベースの設定, 416
国際化機能, 467
ソフトウェア更新, 907
データソースを使用した接続, 18
トラブルシューティング, 1209
トランスポートレイヤーセキュリティを使用する Web サーバーの設定, 1425
トランスポートレイヤーセキュリティを使用するクライアントアプリケーションの設定, 1422
トランスポートレイヤーセキュリティを使用するデータベースサーバーの設定, 1420
認証アプリケーション, 181
SQL Anywhere メトリックリスト, 1265
ローカライズ版, 468
SQL Anywhere JDBC ドライバー
サポート対象外, 435
SQL Anywhere MIB
Agent テーブル, 1334
saDbOptMetaDataTable, 1337
saDbPropMetaDataTable, 1336
saDbStatMetaDataTable, 1336
saMetaData テーブル, 1334
saSrvPropMetaDataTable, 1335
saSrvStatMetaDataTable, 1334
サーバー統計, 1337
サーバープロパティ, 1341
説明, 1321
データベースオプション, 1362
データベース統計, 1350
データベースプロパティ, 1355
テーブルのリスト, 1333
SQL Anywhere MIB リファレンス
概要, 1333
SQL Anywhere ODBC ドライバー
スレッドバージョンと非スレッドバージョン, 45
説明, 36
SQL Anywhere OEM Edition
connection_authentication オプション, 608
database_authentication オプション, 612
アプリケーション認証, 184
アプリケーションの開発, 182
説明, 181
データベース認証, 183
データベースのアップグレード, 186
認証シグネチャー, 182
認証接続, 184
SQL Anywhere SNMP Extension Agent
sasnmp.ini ファイル, 1325
関数の実行, 1330
再起動, 1328
サポートされる MIB, 1321

サポートされるプラットフォーム, 1319
ストアドプロシージャーの実行, 1330
設定, 1324
説明, 1319
動的トラップ, 1331
SQL Anywhere VSS ライター
dbvss12.exe, 1084
-vss dbeng12/dbsrv12 サーバーオプション, 287
SQL Anywhere 環境変数
Mac OS X での設定, 444
SQL Anywhere コンソールユーティリティ (dbconsole)
Mac OS X のハードウェア要件, 906
起動, 905
構文, 1021
使用, 905
設定, 906
ソフトウェア更新, 907
ユーザーとの接続の切断, 543
SQL Anywhere サポートユーティリティ (dbsupport)
SADIAGDIR 環境変数, 449
SQL Anywhere 照合アルゴリズム (SACA)
説明, 488
SQL Anywhere でのスレッド化
説明, 161
SQL Anywhere トランスポートレイヤーセキュリティ
説明, 1410
SQL Anywhere の環境変数
UNIX, 443
UNIX 上のソース, 443
Windows での設定, 443
設定, 443
説明, 443
SQL Anywhere のローカライズ版
説明, 468
SQL Anywhere 12 プラグイン
ER 図, 835
アプリケーションプロファイリングモード, 830
[概要] タブ, 837
使用, 830
設計モード, 830
デバッグモード, 830
SQL Anywhere ボリュームシャドウコピーサービス (VSS)
バックアップの種類, 1084
SQLCONNECT 環境変数
dbstop ユーティリティでの使用, 1026
接続, 30
説明, 455
SQLDA
ansi_blanks オプション, 592
SQL Flagger
sql_flagger_error_level オプション, 679
sql_flagger_warning_level オプション, 680
SQLPATH 環境変数
説明, 456
SQL Remote
SQL Remote レプリケーションオプション, 588
Windows Mobile でサポート対象外の機能, 441
トランザクションログの自動名前変更, 1108
SQLREMOTE 環境変数
説明, 456
SQL 互換性
データベースオプション, 584
SQL コマンドファイル
Interactive SQL で開く, 847
SQL スクリプトファイル
Interactive SQL をデフォルトのエディターにする, 851
SQL 標準
Transact-SQL 互換性オプション, 585
UPDATE 文, 985
SQL ファイルフォーマット
Interactive SQL 出力, 898
Interactive SQL デフォルトエディター, 847
ファイルのバックアップ, 848
ファイルをお気に入りに追加, 855
SQL 文
Interactive SQL でのロギング, 856
Sybase Central でのロギング, 154
Windows Mobile でサポート対象外の文, 436
クライアント／サーバー, 479
最後の作成の取得, 297
データベースオプション, 571
ユーティリティデータベース, 135
[SQL 文] ウィンドウ枠
説明, 843
SQL 文の実行
Interactive SQL, 848, 849
SQL 文のロギング
Interactive SQL, 856
説明, 154
sr_date_format オプション

SQL Remote のオプション, 681
sr_time_format オプション
SQL Remote のオプション, 682
sr_timestamp_format オプション
SQL Remote のオプション, 683
sr_timestamp_with_time_zone_format オプション
SQL Remote のオプション, 684
SSL (セキュアソケットレイヤー)
説明, 1410
SSPI
SQL Anywhere クライアントでサポートされている, 62
SQL Anywhere サーバーでサポートされていない, 62
Windows での Kerberos ログイン, 68
st_geometry_asbinary_format オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 686
st_geometry_asbinary_format プロパティ
接続プロパティの説明, 713
st_geometry_astext_format オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 687
st_geometry_astext_format プロパティ
接続プロパティの説明, 713
st_geometry_asxml_format オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 688
st_geometry_asxml_format プロパティ
接続プロパティの説明, 713
st_geometry_describe_type オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 689
st_geometry_describe_type プロパティ
接続プロパティの説明, 713
st_geometry_interpolation オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 690
st_geometry_interpolation プロパティ
接続プロパティの説明, 713
st_geometry_on_invalid オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 691
st_geometry_on_invalid プロパティ
接続プロパティの説明, 713
StartDBPermission プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
START JAVA 文
Windows Mobile でサポート対象外, 436
StartLine 接続パラメーター
組み込みデータベース, 28
説明, 366
StartTime プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
START 接続パラメーター
組み込みデータベース, 28
説明, 366
StatementDescribes プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
StatementPostAnnotatesSimple プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
StatementPostAnnotatesSkipped プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
StatementPostAnnotates プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
STOP JAVA 文
Windows Mobile でサポート対象外, 436
StreamsUsed プロパティ
サーバープロパティの説明, 754
string_rtruncation オプション
SQL Anywhere SNMP Extension Agent OID, 1362
Transact-SQL 互換性オプション, 585
説明, 691
string_rtruncation プロパティ
接続プロパティの説明, 713
subscribe_by_remote オプション
SQL Remote レプリケーションオプション, 588
説明, 692
subsume_row_locks オプション
SQL Anywhere SNMP Extension Agent OID, 1362

説明, 693
subsume_row_locks プロパティ
接続プロパティの説明, 713
suppress_tds_debugging オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 693
suppress_tds_debugging プロパティ
接続プロパティの説明, 713
SuppressWarnings 接続パラメーター
ODBC 接続パラメーターの説明, 942
Sybase Central
Mac OS X のハードウェア要件, 817
SQL Anywhere 12 プラグイン, 830
Windows Mobile, 427
Windows Mobile でサポート対象外のウィザード, 439
監査, 1388
起動, 816
キーボードショートカット, 823
高速ランチャーの設定, 903
コードエディター, 824
ステータスバー, 822
接続プロファイル, 9
説明, 815
ソフトウェア更新, 907
テキスト補完, 901
データベースオブジェクトのコピー, 831
データベースの削除, 140
ナビゲーション, 820
右ウィンドウ枠でのカラムのカスタマイズ, 822
レジストリ設定, 466
ログビューアー, 829
Sybase IQ
Interactive SQL との接続, 841
SYBASE-MIB.mib
説明, 1320
Sybase Open Client
ASE 互換性オプション, 585
Kerberos 認証, 67
Windows Mobile でサポート対象外, 435
インターフェイス, 1435
オプション, 1443
クライアント API と文字セット変換, 480
設定, 1438
データベース互換性オプション, 585
Sybase Open Server
JDBC のサーバーの設定, 1443
アドレス, 1440
アーキテクチャー, 1435
起動, 1437
サーバーエントリの削除, 1442
サーバーエントリ名の変更, 1442
システムの稼働条件, 1436
追加, 1438
Sybase PowerBuilder DataWindow
クエリパフォーマンス, 655
SYBASE 環境変数
DSEdit, 1439
説明, 456
sybinit ユーティリティ
説明, 1437
SynchronizationSchemaChangeActive プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
synchronize_mirror_on_commit オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 694
synchronize_mirror_on_commit プロパティ
接続プロパティの説明, 713
SyncTrunc プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
SYS_SPATIAL_ADMIN_ROLE グループ
特殊なグループとユーザー, 551
SYSCOLAUTH
統合ビューに対するパーミッション, 519
SYSCOLPERM
システムビューに対するパーミッション, 519
SYSGROUP
システムビューに対するパーミッション, 519
SYSGROUPS
統合ビューに対するパーミッション, 519
Syslog
ユーザー ID, 268
SYSPROCAUTH
統合ビューに対するパーミッション, 519
SYSPROCPERM
システムビューに対するパーミッション, 519
SYSTABAUTH
統合ビューに対するパーミッション, 519
SYSTABLEPERM

システムビューに対するパーミッション, 519
system DB 領域
説明, 120
SYSUSER
統合ビューに対するパーミッション, 519
SYSUSERAUTHORITY
システムビューに対するパーミッション, 519
SYSUSERLIST
統合ビューに対するパーミッション, 519
SYSUSERPERM
互換ビューに対するパーミッション, 519
SYSUSERPERMS
互換ビューに対するパーミッション, 519
SYS グループ
説明, 551

## T

Tabular Data Stream 通信プロトコル
Sybase Open Server, 1435
TCP/IP
BroadcastListener (BLISTENER) プロトコルオプション, 373
ClientPort (CPORT) プロトコルオプション, 377
Host (IP) プロトコルオプション, 381
IPv6 サポート, 100
LDAP プロトコルオプション, 387
Open Server アドレス, 1440
ServerPort (PORT) プロトコルオプション, 396
SQL Anywhere クライアントの Mobile Link TLS, 1431
Ultra Light クライアントの Mobile Link TLS, 1433
Windows, 100
-x サーバーオプション, 289
起動, 97
クライアント／サーバー通信の暗号化, 100
サポートされているプロトコル, 96
サーバー設定, 369
サービス依存性の設定, 1016
説明, 99
データベースミラーリングに必要, 1140
パフォーマンス, 95
ファイアウォール経由のサーバー検索, 1002
ファイアウォール経由の接続, 101
ファイアウォールと LDAP サーバー, 103
プロトコルオプション, 370
ポートの識別, 396
ポート番号, 398
TcpIpAddresses プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
TDI
サービス依存性の設定, 1016
tds_empty_string_is_null オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 694
tds_empty_string_is_null プロパティ
接続プロパティの説明, 713
TDS 通信プロトコル
説明, 1435
TDS プロトコルオプション
説明, 399
temp_space_limit_check オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 695
temp_space_limit_check プロパティ
接続プロパティの説明, 713
temp DB 領域
説明, 120
TempDir プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
TempDiskSpace システムイベント
説明, 1127
TempFileName プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
TempFilePages プロパティ
接続プロパティの説明, 713
temporary DB 領域
説明, 120
パーミッション, 122
TempTablePages 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
TempTablePages プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
TEMP 環境変数
Windows Mobile, 466

説明, 457
ディスク領域, 1211
ThreadDeadlocksAvoided 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
ThreadDeadlocksAvoided プロパティ
サーバープロパティの説明, 754
ThreadDeadlocksReported 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
ThreadDeadlocksReported プロパティ
サーバープロパティの説明, 754
time_format オプション
ASE 互換性オプション, 585
SQL Anywhere SNMP Extension Agent OID, 1362
Sybase Open Client, 1443
Transact-SQL 互換性オプション, 585
説明, 696
time_format プロパティ
接続プロパティの説明, 713
time_zone_adjustment オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 697
time_zone_adjustment プロパティ
接続プロパティの説明, 713
Timeout プロトコルオプション
説明, 400
timestamp_format オプション
ASE 互換性オプション, 585
SQL Anywhere SNMP Extension Agent OID, 1362
Sybase Open Client, 1443
Transact-SQL 互換性オプション, 585
説明, 698
timestamp_format プロパティ
接続プロパティの説明, 713
timestamp_with_time_zone_format オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 700
timestamp_with_time_zone_format プロパティ
接続プロパティの説明, 713
TIMESTAMP データ型
default_timestamp_increment オプション, 618
比較する小数点以下の桁数, 619
TimeZoneAdjustment プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
接続プロパティの説明, 713
TLS
Mobile Link クライアント (SQL Anywhere), 1431
Mobile Link クライアント (Ultra Light), 1433
Mobile Link でのエンドツーエンド暗号化, 1426
Windows Mobile でサポート対象外, 438
サポート, 1411
説明, 1410
プロトコルオプション, 370
モニター, 1316
tls_type プロトコルオプション
dbeng12 -ec, 223
dbsrv12 -ec, 223
TLS サポート
説明, 1411
TLS 同期
説明, 1410
TMPDIR 環境変数
Windows Mobile, 466
説明, 457
TMP 環境変数
Windows Mobile, 466
説明, 457
TotalBuffers 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
TotalBuffers プロパティ
サーバープロパティの説明, 754
TO プロトコルオプション
説明, 400
tracetime.pl
要求ログファイルの分析, 642
TransactionStartTime プロパティ
接続プロパティの説明, 713
Transact-SQL
allow_nulls_by_default オプション, 589
NULL の動作, 597
quoted_identifier オプション, 668
カラム NULL の互換性, 668
クエリエディターでサポートなし, 866
更新パーミッション, 594
互換性オプション, 585
削除パーミッション, 594

データベースの互換性オプション, 584
TranslationDLL 接続パラメーター
ODBC 接続パラメーターの説明, 942
TranslationName 接続パラメーター
ODBC 接続パラメーターの説明, 942
TranslationOption 接続パラメーター
ODBC 接続パラメーターの説明, 942
translog DB 領域
説明, 120
translogmirror DB 領域
説明, 120
TriggerPages 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
TriggerPages プロパティ
データベースプロパティの説明, 782
truncate_timestamp_values オプション
Mobile Link 同期の使用, 702
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 702
truncate_timestamp_values プロパティ
接続プロパティの説明, 713
truncation_length オプション
Interactive SQL 設定, 884
説明, 900
trusted_certificate プロトコルオプション
Mobile Link トランスポートレイヤーセキュリティ, 1430
説明, 401
tsql_outer_joins オプション
ASE 互換性オプション, 585
SQL Anywhere SNMP Extension Agent OID, 1362
Transact-SQL 互換性オプション, 585
説明, 703
tsql_outer_joins プロパティ
接続プロパティの説明, 713
tsql_variables オプション
ASE 互換性オプション, 585
SQL Anywhere SNMP Extension Agent OID, 1362
Sybase Open Client, 1443
Transact-SQL 互換性オプション, 585
説明, 703
tsql_variables プロパティ
接続プロパティの説明, 713
Type 2 JDBC
クライアント API と文字セット変換, 480

## U

UAC
Vista での SQL Anywhere の実行, 187
Windows 7 での SQL Anywhere の実行, 187
Windows Server 2008 での SQL Anywhere の実行, 187
UCA
説明, 489
UCA 照合
Swedish Academy 標準, 496
カタカナ, 496
スウェーデン語のロケール, 496
ひらがな, 496
UDP
説明, 99
UID 接続パラメーター
説明, 368
UI の初期化失敗 (タイプ 4)
Linux での UI の表示, 286
ulcond12 ユーティリティ
Vista での権限の昇格の必要性, 188
Windows 7 での権限の昇格の必要性, 188
Windows Server 2008 での権限の昇格の必要性, 188
Ultra Light
Mobile Link トランスポートレイヤーセキュリティ, 1433
Ultra Light クライアント
TLS, 1433
UncommitOp プロパティ
接続プロパティの説明, 713
Unconditional 接続パラメーター
説明, 368
UNC 接続パラメーター
説明, 368
Unicode 照合アルゴリズム (UCA)
説明, 489
Unicode 文字セット
説明, 487
UniqueClientAddresses 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
UniqueClientAddresses プロパティ
サーバープロパティの説明, 754
UNIX

dbconsole の起動, 906
Interactive SQL の起動, 842
IPv6 サポート, 100
LD_LIBRARY_PATH 環境変数, 445
ODBC_INI 環境変数, 447
ODBCHOME 環境変数, 447
ODBCINI 環境変数, 447
ODBC サポート, 48
SATMP 環境変数, 457
Sybase Central の起動, 818
環境変数の設定, 443
キャッシュサイズ, 204
サンプルデータベースへの接続, 7
システム情報ファイル, 48
照合の選択, 508
スレッド接続ライブラリと Ping ユーティリティの使用, 994
スレッドの動作, 163
デフォルトの文字セット, 487
テンポラリファイル, 452
テンポラリファイルパーミッション, 452
データベースサーバーの起動, 145
パフォーマンス統計値のモニタリング, 991
ファイルの検索, 462
ファイルのソース指定, 444
保護された共有メモリ接続, 1381
ライセンス実行プログラム, 1006
UNLOAD TABLE 文
WRITECLIENTFILE 権限, 541
セキュリティ, 1396
UNLOAD 文
セキュリティ, 1396
unprocessed.sql
説明, 1060
UnschReq 統計情報
SQL Anywhere SNMP Extension Agent OID, 1337
UnschReq プロパティ
サーバープロパティの説明, 754
updatable_statement_isolation オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 704
updatable_statement_isolation プロパティ
接続プロパティの説明, 713
update_statistics オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 705
update_statistics プロパティ
接続プロパティの説明, 713
update_timeout パラメーター
LDAP, 105
UPDATE パーミッション
説明, 520
付与, 526
UPDATE 文
Interactive SQL での生成, 859
文字列のトランケーション, 691
upgrade_database_capability オプション
SQL Anywhere SNMP Extension Agent OID, 1362
upgrade_database_capability プロパティ
接続プロパティの説明, 713
user_estimates オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 706
user_estimates プロパティ
接続プロパティの説明, 713
UserAppInfo プロパティ
接続プロパティの説明, 713
UserDefinedCounterRate01 プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
UserDefinedCounterRate02 プロパティ
サーバープロパティの説明, 754
UserDefinedCounterRate02 プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
UserDefinedCounterRate03 プロパティ
サーバープロパティの説明, 754
UserDefinedCounterRate03 プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
UserDefinedCounterRate04 プロパティ
サーバープロパティの説明, 754
UserDefinedCounterRate04 プロパティ
接続プロパティの説明, 713
データベースプロパティの説明, 782
UserDefinedCounterRate05 プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
UserDefinedCounterRaw01 プロパティ

サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
UserDefinedCounterRaw02 プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
UserDefinedCounterRaw03 プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
UserDefinedCounterRaw04 プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
UserDefinedCounterRaw05 プロパティ
サーバープロパティの説明, 754
接続プロパティの説明, 713
データベースプロパティの説明, 782
Userid 接続パラメーター
説明, 368
UserID プロパティ
接続プロパティの説明, 713
UTF-8
データベースでの UTF-8 の使用, 959
util_db.ini
説明, 139
UtilCmdsPermitted プロパティ
接続プロパティの説明, 713
utility_db
接続, 137
文実行パーミッションの制御, 561
ユーティリティデータベースに予約された名前, 135
uuid_has_hyphens オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 707
uuid_has_hyphens プロパティ
接続プロパティの説明, 713

## V

VALIDATE 権限
継承不可能, 541
説明, 541
付与, 542
VCS エージェント
SADatabase, 1186
SAServer, 1184
verify_all_columns オプション
SQL Remote レプリケーションオプション, 588
説明, 707
verify_password_function オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 708
パスワードの認証, 1384
verify_password_function プロパティ
接続プロパティの説明, 713
verify_threshold オプション
SQL Remote レプリケーションオプション, 588
説明, 711
VerifyServerName プロトコルオプション
説明, 402
VERIFY プロトコルオプション
説明, 402
Veritas Cluster Server
SQL Anywhere での使用, 1182
Veritas Cluster Server エージェント
説明, 1182
VersionStorePages 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
VersionStorePages プロパティ
データベースプロパティの説明, 782
viewcert ユーティリティ
構文, 934
使用法, 1414
ViewPages 統計情報
SQL Anywhere SNMP Extension Agent OID, 1350
ViewPages プロパティ
データベースプロパティの説明, 782
Vista
ActiveSync, 188
AWE キャッシュの使用, 188
dbconsole ユーティリティを使用したデータベースサーバーのモニタリング, 188
dbelevate12.exe, 188
DisableMultiRowFetch 接続パラメーター, 340
SQL Anywhere 昇格操作エージェント, 188
SQL Anywhere の実行, 187
Windows Mobile Device Center, 188
サービスがデスクトップと対話しない, 188
署名された実行プログラム, 188

配備に関する考慮事項, 188
ユーザーアカウント制御, 188
VSS
SQL Anywhere ボリュームシャドウコピーサービス, 1084
データベースサーバー, 287
VSS ライター
dbvss12.exe, 1084


## W


wait_for_commit オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 711
wait_for_commit プロパティ
接続プロパティの説明, 713
WaitStartTime プロパティ
接続プロパティの説明, 713
WaitType プロパティ
接続プロパティの説明, 713
WebClientLogFile プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
WebClientLogging プロパティ
SQL Anywhere SNMP Extension Agent OID, 1341
サーバープロパティの説明, 754
webservice_namespace_host オプション
SQL Anywhere SNMP Extension Agent OID, 1362
説明, 712
webservice_namespace_host プロパティ
接続プロパティの説明, 713
Web サーバー
トランスポートレイヤーセキュリティを使用する起動, 1425
ミラーリングシステムでの使用, 1141
ログファイルサイズ, 390
ログファイルのメッセージフォーマット, 389
Web サービス
webservice_namespace_host オプション, 712
デフォルトで実行されるログインプロシージャー, 637
データベースサーバー設定, 294
トランスポートレイヤーセキュリティを使用する起動, 1425
Web サービスクライアント
certificate_company プロトコルオプション, 374
certificate_name プロトコルオプション, 376
certificate_unit プロトコルオプション, 377
trusted_certificates プロトコルオプション, 401
ロギング, 300
Web サービスクライアントログファイル
名前の設定, 300
Windows
AWE キャッシュサイズの制限, 217
IPv6 サポート, 100
SNMP のインストール, 1325
Sybase Central の起動, 816
TCP/IP, 100
イベントログ, 155
キャッシュサイズ, 204
コードページ, 476
最小キャッシュサイズ, 211
サービス, 171
照合の選択, 507
スレッドの動作, 163
デフォルトの文字セット, 487
データベースサーバーの起動, 145
統合化ログイン, 51
ファイルの検索, 461
文字セット, 476
レジストリ設定のインストール, 466
Windows 2003
SNMP のインストール, 1325
Windows 7
ActiveSync, 188
AWE キャッシュの使用, 188
dbconsole ユーティリティを使用したデータベースサーバーのモニタリング, 188
dbelevate12.exe, 188
SQL Anywhere 昇格操作エージェント, 188
SQL Anywhere の実行, 187
Windows Mobile Device Center, 188
サービスがデスクトップと対話しない, 188
署名された実行プログラム, 188
配備に関する考慮事項, 188
ユーザーアカウント制御, 188
Windows CE
IPv6 アドレス−サポートされない, 100
Windows MIT Kerberos クライアント
keytab ファイル, 63
Windows Mobile
.NET Compact Framework の使用, 405

@data オプションはサポート対象外, 438
ADO.NET Sample の使用, 408
dbxtract はサポート対象外, 441
ESQL Sample, 409
ICU, 404
Interactive SQL からのデータベースの管理, 432
IP アドレス, 415
jConnect, 417
Kerberos はサポート対象外, 435
LDAP はサポート対象外, 435
ODBC Sample, 410
ODBC データソースの作成, 412
ODBC データソースの使用, 47
SQL Anywhere JDBC ドライバーはサポート対象外, 435
SQL Anywhere インストール要件, 404
SQL Anywhere の使用, 403
Sybase Open Client はサポート対象外, 435
TEMP ファイルの設定, 466
Windows Mobile 5.0 for Smartphone の制限, 406
アプリケーションプロファイリングの制限, 435
暗号化, 418
インストール, 459
外部環境はサポート対象外, 435
外部ストアドプロシージャーはサポート対象外, 435
監査, 1409
管理ユーティリティ, 427
サブフォルダー, 459
サポートされる機能, 434
サポート対象外の SQL Anywhere の機能, 435
サポート対象外の SQL Remote の機能, 441
サポート対象外の SQL 文, 436
サポート対象外の Sybase Central ウィザード, 439
サポート対象外の管理ツール, 435
サポート対象外のデータベースサーバーオプション, 437
サンプル, 406
[サーバー起動オプション] ウィンドウ, 427
サーバーの停止, 431
照合の適合化のサポートの制限, 416
ストレージカード, 404
制限される jConnect 機能, 435
制限される ODBC 機能, 435
セキュリティ, 1408
接続プールはサポート対象外, 435
チェックサムがデフォルトで有効, 1121
抽出ユーティリティはサポート対象外, 441
通信の暗号化, 1409
ディレクトリアクセスサーバーはサポート対象外, 435
デスクトップからの接続, 36
デスクトップコンピューターからの接続, 411
デバイスセキュリティ, 1408
デバイスへのデータベースのコピー, 422
データベースサーバー, 426
データベースサーバーオプション, 1408
データベースサーバーの起動, 411
データベースの暗号化, 1409
データベースの再構築, 423
データベースの作成, 418
データベースの消去, 426
データベースの設定, 416
データベースのバックアップ, 425
データベースミラーリングはサポート対象外, 435
トランザクションログ, 417
パーソナルサーバーはサポート対象外, 435
ファイルの検索, 462
ファイルのロケーション, 459
複数のデータベースの開始, 429
プロキシポート, 413
並列バックアップはサポート対象外, 435
リモートデータアクセスはサポート対象外, 435
Windows Mobile 5.0
スマートフォンの制限, 406
Windows Mobile Device Center
必要なバージョン, 404
Windows Mobile のデータベースの再構築
説明, 423
[Windows Mobile プロキシポート] ウィンドウ
使用, 414
Windows Mobile 用データベースの作成
CREATE DATABASE 文, 421
dbinit ユーティリティ, 421
Interactive SQL, 421
Sybase Central, 419
Windows Server 2008
ActiveSync, 188
AWE キャッシュの使用, 188
dbconsole ユーティリティを使用したデータベースサーバーのモニタリング, 188
dbelevate12.exe, 188

SQL Anywhere 昇格操作エージェント, 188
Windows Mobile Device Center, 188
サービスがデスクトップと対話しない, 188
署名された実行プログラム, 188
配備に関する考慮事項, 188
ユーザーアカウント制御, 188
Windows Vista
DisableMultiRowFetch 接続パラメーター, 340
SNMP のインストール, 1325
Windows XP
SNMP のインストール, 1325
統合化ログイン, 51
Windows サービス
Windows サービス マネージャー, 178
アカウントオプション, 175
新しいデータベースの追加, 176
依存性, 179
オプション, 175
開始, 177
概要, 171
起動オプション, 174
起動順序, 179
グループ, 178
再表示, 177
削除, 173
作成, 172
実行ファイル, 176
設定, 174
停止, 177
適格なプログラム, 171
デスクトップのアイコン, 175
データベースサーバー, 170
パラメーター, 174
複数, 179
レジストリ設定, 464
Windows サービスマネージャー
説明, 178
Windows パフォーマンスモニター
共有メモリ作成の無効化, 254
モニター対象の接続数の制御, 255
モニター対象のデータベース数の制御, 255
Windows ユーザーグループ
統合化ログイン, 57
Winsock
Windows での TCP/IP の使用, 100
WITH GRANT OPTION 句
使用, 531
WriteChecksum プロパティ
SQL Anywhere SNMP Extension Agent OID, 1355
データベースプロパティの説明, 782
WRITECLIENTFILE 権限
継承可能, 541
説明, 541
付与, 542

## X

X.509 証明書
作成, 924
表示, 934
XML
Interactive SQL での表示, 871
XML ファイルフォーマット
Interactive SQL 出力, 898
xp_cmdshell システムプロシージャー
セキュリティ機能, 1380
xp_sendmail システムプロシージャー
セキュリティ機能, 1380
xp_srvmon_count_unsubmitted_crash_reports プロシージャー
モニター, 1311
xp_startmail システムプロシージャー
セキュリティ機能, 1380
xp_startsmtp システムプロシージャー
セキュリティ機能, 1380
xp_stopmail システムプロシージャー
セキュリティ機能, 1380
xp_stopsmtp システムプロシージャー
セキュリティ機能, 1380
XPathCompiles プロパティ
データベースプロパティの説明, 782
X Window Server
Linux での UI の表示, 286

## あ

アイコン
サービス実行用, 175
アイドル状態のサーバー
イベントの例, 1130
アクセスプラン
オプティマイザーによる使用を制御, 657
アクセント記号の区別
データベース, 959
フランス語の規則の使用, 959
アクティブなタスク

説明, 165
アクティブな要求
説明, 165
値
Interactive SQL での編集, 860
圧縮
暗号化されたデータベースファイル, 1396
カラム, 117
パケット, 263
パフォーマンス, 106
圧縮されたカラム
説明, 117
アップグレード
データベース, 1061
認証データベース, 186
アップグレードユーティリティ (dbupgrad)
構文, 1061
終了コード, 1064
アプリケーション
SQL Anywhere OEM Edition, 181
アプリケーションの開発
説明, 182
アプリケーションの認証
説明, 184
アプリケーションプロファイリング
Windows Mobile での制限, 435
アプリケーションプロファイリングモード
説明, 830
暗号
トランスポートレイヤーセキュリティ, 1410
暗号化
AES アルゴリズム, 1397
certificate_company プロトコルオプション, 374
certificate_name プロトコルオプション, 375
certificate_unit プロトコルオプション, 376
CREATE ENCRYPTED FILE と CREATE ENCRYPTED DATABASE の比較, 1400
dbinit を使用したデータベースの作成, 958
-ec サーバーオプション, 223
-ek サーバーオプション, 310
EncryptedPassword (ENP) 接続パラメーター, 341
Encryption (ENC) 接続パラメーター, 342
-ep サーバーオプション, 226
-es サーバーオプション, 227
FIPS, 1412
INI ファイル, 953
Mobile Link, 1426
TDS プロトコルオプション, 399
trusted_certificate プロトコルオプション, 401
Windows Mobile, 418
Windows Mobile 上でのクライアント／サーバー通信, 1409
Windows Mobile 上の SQL Anywhere データベース, 1409
暗号化されたデータベースのパフォーマンス, 1403
エンドツーエンド, 1426
カラム, 1403
強力, 1397
説明, 1396
単純, 1396
通信, 1410
テクニカルサポートのためのデータベース, 1400
データベース, 1398
データベースの復号化, 1401
データベースファイル, 1396
データベースファイルの作成後, 1400
テーブル, 1405, 1406
テーブルの作成後, 1407
テーブルの作成時, 1407
パスワード, 1383
ファイル非表示ユーティリティ (dbfhide), 953
有効化, 1406
暗号化アルゴリズム
AES, 1397
Rijndael, 1397
暗号化キー
CREATE ENCRYPTED DATABASE 文を使用した変更, 1402
DBKEY 接続パラメーター, 335
暗号化されたデータベースの起動, 335
アンロードユーティリティ (dbunload), 1044
指定, 959
使用, 1396
消去ユーティリティ (dberase), 951
初期化ユーティリティ (dbinit), 959
選択, 1402
トランザクションログユーティリティ (dblog), 1041
保護, 1402
ログ変換ユーティリティ (dbtran), 985
暗号化接続パラメーター
クライアント／サーバー通信の保護, 1422
暗号化プロパティ

接続プロパティの説明, 713
暗号方式
パブリックキー, 1410
安全なデータの管理
概要, 1379
アンパサンド記号
設定ファイルでの使用, 909
アンロード
セキュリティ, 1396
トラブルシューティング, 1060
トラブルシューティング
アンロードの失敗, 1060
アンロードユーティリティ (dbunload)
DB 領域ファイル名, 1044
構文, 1044
終了コード, 1058
アーカイブ
テープに直接バックアップ, 1083
アーカイブバックアップ
説明, 1077
定義, 1076
リストア, 1091


## い


移行
Relay Server モニター, 1314
依存性
サービス, 179
サービス依存性の管理, 179
設定, 1016
イベント
イベントハンドラーを隠す, 1136
システム, 1124
手動, 1124
手動トリガーイベントの作成, 1134
手動のトリガー, 1135
条件, 1124
処理, 1123
スケジュール, 1124
スケジュール作成ウィザード, 1126
説明, 1124
定義, 1123, 1126
データベースドキュメントの生成, 838
データベースミラーリング, 1140
内部, 1131
イベント作成ウィザード
使用, 1133
イベント処理
説明, 1123
イベントスケジュール
定義, 1126
イベントタイプ
説明, 1132
イベントの処理
説明, 1123
イベントハンドラー
定義, 1123
デバッグ, 1131
トランザクションの動作, 1133
内部, 1133
ライセンスされている接続への影響, 1133
イベントハンドラー
隠す, 1136
イベントログ
メッセージを出力しない, 155
イメージ
Interactive SQL での表示, 869
イメージのバックアップ
dbbackup からのエラーメッセージの受信, 913
バックアップユーティリティ (dbbackup) を使用して作成, 913
バックアップユーティリティ (dbbackup) を使用して実行, 913
イメージバックアップ
説明, 1077
定義, 1076
並列, 1116
元のトランザクションログの名前を変更, 1110
リストア, 1091
インクリメンタルバックアップ
説明, 1074
バックアップ ユーティリティ (dbbackup), 1082
印刷
Interactive SQL, 872
インストール
Windows Mobile, 459
Windows Mobile への SQL Anywhere のインストール, 404
モニターを別のコンピューターにインストール, 1312
レジストリ設定, 466
ロケーション, 458
インストールされたオブジェクト
モニター、修復, 1260
インストールディレクトリ

SQLANY12, 453
説明, 458
インターフェイス識別子
IPv6 アドレス, 101
Linux で必要, 101
インターフェイス名
IPv6 アドレス, 101
インターフェイスライブラリ
検出, 77
接続, 5
インデックス
制限, 811
インデント
Interactive SQL のショートカット, 878
Interactive SQL の説明, 850
インポート
MobiLink リソースのインポート, 1252
Relay Server リソースのインポート, 1252
接続プロファイル, 13
モニター、リソースのインポート, 1252
インメモリモード
設定, 247
引用符
接続文字列での使用, 31

## う

ウィザード
Windows Mobile でサポート対象外の Sybase Central ウィザード, 439
ウィジェット
モニター, 1244
ウォッチリスト
データベースオプション, 573
後ろのスペース
接続文字列での使用, 31

## え

英語以外のデータベース
作成, 503
エクスポート
接続プロファイル, 14
エスケープ文字
ODBC データソース, 942
アンロードユーティリティ (dbunload), 1044
設定ファイル, 910
データベースの作成, 118
エディション
SQL Anywhere OEM Edition, 181
エラー
Interactive SQL, 857
Transact-SQL プロシージャー, 655
イベントハンドラーの動作, 1133
レポートの iAnywhere Solutions への送信, 1210
エラー処理
Interactive SQL, 898
Transact-SQL プロシージャー, 655
エラーのレポート
説明, 1210
エラーレポート
サポートユーティリティ (dbsupport), 1027
説明, 1210
エンコード
Mac OS X の推奨, 506
説明, 473
エンタープライズルート証明書
作成, 932
証明書チェーン, 1415
トランスポートレイヤーセキュリティ, 1417
エンタープライズルート証明書の作成
トランスポートレイヤーセキュリティ, 1417
エンドツーエンド暗号化
createkey ユーティリティ, 981
Mobile Link、説明, 1426
エンリスト
再エンリストリカバリ時間, 280
エージェント
説明, 1320

## お

大文字と小文字の区別
dbinit ユーティリティ, 958
コマンドライン, 199
サーバー名, 152
初期化ユーティリティ (dbinit), 959
接続パラメーター, 321
データベースオプション, 569
データベース名, 152
トルコ語の大文字と小文字を区別しないデータベース, 511
トルコ語の大文字と小文字を区別するデータベース, 509
プロトコルオプション, 370
大文字と小文字を区別する
国際的な側面, 480

お気に入りに追加
Interactive SQL の説明, 855
お気に入りリスト
インポート, 856
エクスポート, 856
共有, 856
説明, 854
追加, 855
表示, 855
オブジェクト
修飾された名前, 562
オプション
ASE 互換性オプション, 585
Broadcast Repeater ユーティリティ (dbns12), 921
dbisqlc ユーティリティ, 950
Interactive SQL オプション, 884
Interactive SQL ユーティリティ (dbisql), 976
isql_allow_read_client_file, 890
isql_print_result_set, 895
Linux サービスユーティリティ (dbsvc), 1007
ping ユーティリティ (dbping), 992
PUBLIC オプション, 570
SQL Anywhere MIB 内のデータベースオプション, 1362
SQL Anywhere Mobile Link クライアント用の設定, 587
SQL Anywhere コンソールユーティリティ (dbconsole), 1021
SQL Anywhere スクリプト実行ユーティリティ (dbrunsql), 997
SQL Remote レプリケーションオプション, 588
sr_date_format, 681
Sybase Open Client, 1443
Transact-SQL 互換性オプション, 585
Windows サービスユーティリティ (dbsvc), 1013
値の検索, 571
アップグレードユーティリティ (dbupgrad), 1062
アンロードユーティリティ (dbunload), 1044
大文字と小文字の区別, 569
起動時の設定, 1444
言語選択ユーティリティ (dblang), 983
検証ユーティリティ (dbvalid), 1064
サポートユーティリティ (dbsupport), 1027
サーバー停止ユーティリティ (dbstop), 1024
サーバーバックグラウンド起動ユーティリティ (dbspawn), 1023
サーバーライセンス取得ユーティリティ (dblic), 1003
サーバー列挙ユーティリティ (dblocate), 999
消去ユーティリティ (dberase), 951
照合の適合化, 496
情報ユーティリティ (dbinfo), 957
証明書作成ユーティリティ (createcert), 924
証明書ビューアーユーティリティ (viewcert), 934
初期化ユーティリティ (dbinit), 959
設定の削除, 574
設定のモニタリング, 573
説明, 568
テンポラリの設定, 570
データソースユーティリティ (dbdsn), 936
データベースオプションのアルファベット順リスト, 575
データベースオプションの初期設定, 574
データベースオプションのスコープと継続期間, 569
データベースオプションの設定, 568
データベースサーバー (dbeng12/dbsrv12), 189
トランザクションログユーティリティ (dblog), 1041
バックアップユーティリティ (dbbackup), 913
ヒストグラムユーティリティ (dbhist), 955
分類, 575
ユーザーオプションとグループオプションの設定, 516
ログ変換ユーティリティ (dbtran), 985
[オプション] ウィンドウ
Interactive SQL, 846
オプションのウォッチリスト
説明, 573
オプティマイザー
アクセスプランの検索に使用される作業量の制御, 657
バイパス, 657
オフライントランザクションログ
バックアップ, 1116
オフラインバックアップ
説明, 1072
オペレーター
モニターのユーザー, 1297
オンラインバックアップ
説明, 1072

## か


会社名
　サーバーライセンス取得ユーティリティ (dblic), 1003
階層
　コピーノードの移動, 1200
外部アンロード
　使用, 1058
外部環境
　Windows Mobile でサポート対象外, 435
外部関数
　スタックサイズ, 234
外部ストアドプロシージャー
　Windows Mobile でサポート対象外, 435
外部ログイン
　ユーザーの削除時に削除, 517
[概要] タブ
　SQL Anywhere, 837
書き込みチェックサム
　I/O 操作, 1120
　-wc サーバーオプション, 288
　検証ユーティリティ (dbvalid), 1064
　自動, 1121
　説明, 1120
拡張文字
　説明, 474
カタカナ
　照合の適合化, 496
活性
　接続, 278
稼働条件
　Veritas Cluster Server エージェント, 1183
可変幅の文字セット
　説明, 475
下方コードページ
　説明, 474
可用性
　高い, 1075
　データベースサーバー, 170
カラム
　Interactive SQL 内の検索, 857
　NULL 値の許可, 115
　圧縮, 117
　暗号化, 1403
　制限, 811
　制約, 117
　データ型の選択, 115
　パーミッション, 526
　命名, 114
カラム統計
　collect_statistics_on_dml_updates オプション, 606
カラムの圧縮
　説明, 117
カラムのコピー
　Interactive SQL, 862
カラムパーミッション
　設定, 526
カラム名
　国際的な側面, 480
環境変数
　@data オプション, 909
　DYLD_LIBRARY_PATH, 445
　ERRORLEVEL, 976
　LD_LIBRARY_PATH, 445
　LIBPATH, 446
　Mac OS X での設定, 444
　ODBC_INI, 447
　ODBCHOME, 447
　ODBCINI, 447
　PATH, 447
　SACHARSET, 449
　SADIAGDIR, 449
　SALANG, 450
　SALOGDIR, 450
　SATMP, 451
　SHLIB_PATH, 453
　SQLANY12, 453
　SQLANYSAMP12, 454
　SQLCONNECT, 455
　SQLPATH, 456
　SQLREMOTE, 456
　SYBASE, 456
　TEMP, 457
　TMP, 457
　TMPDIR, 457
　UNIX, 443
　UNIX 上のソース, 443
　Windows Mobile での TEMP ディレクトリの設定, 466
　Windows での設定, 443
　設定, 443
　説明, 443
　データベースユーティリティへの接続, 29
監査

conn_auditing オプション, 607
Sybase Central, 1388
Windows Mobile 上のデータベース, 1409
監査情報の取り出し, 1390
コミットされない操作のリカバリ, 1090
コメント, 1392
制御, 598
セキュリティ機能, 1379
接続, 1390
説明, 1388
トランザクションログユーティリティ (dblog) の操作, 1395
無効化, 1388
有効化, 1388
例, 1392
ログ変換ユーティリティ (dbtran), 985
ログ変換ユーティリティ (dbtran) の操作, 1395
監視サーバー
コピーノードーの使用, 1167
接続文字列の指定, 290
停止, 1168
データベースミラーリングシステムのロール, 1142
データベースミラーリングの概要, 1137
データベース名の指定, 290
監視サーバー
削除, 1169
関数
SQL Anywhere SNMP Extension Agent による実行, 1330
データベースドキュメントの生成, 838
カンマ区切りファイル
output_format オプション, 898
カンマ区切りファイルフォーマット
input_format オプション, 890
管理者
モニターのユーザー, 1297
管理ツール
Interactive SQL, 840
Mac OS X のハードウェア要件, 817
Sybase Central, 815
Windows Mobile でサポート対象外の機能, 435
言語設定の変更, 904
管理ユーティリティ
(*参照* データベース管理ユーティリティ)
Windows Mobile, 427
カーソル
ansi_close_cursors_on_rollback オプション, 593
close_on_endtrans オプション, 605
max_cursor_count オプション, 642
接続制限, 552
データベースオプション, 569
トランザクション, 605

## き

記号
結果を表示した場合の予期しない記号のトラブルシューティング, 479
「起動できません。」エラー
原因の診断, 95
キャッシュ
max_plans_cached オプション, 643
Vista での AWE, 188
Windows 7 での AWE, 188
Windows Server 2008 での AWE, 188
最小サイズ, 211
サイズオプション, 156
最大サイズ, 811
サーバー名, 91
データベースサーバーオプション, 156
キャッシュウォーミング
キャッシュとページの再ロード, 214
サーバーメッセージ, 216
データベースページの収集, 207
キャッシュサイズ
AWE キャッシュサイズの制限, 212
暗号化されたデータベースの問題, 1403
最小値, 211
最小値の設定, 211
最大値の設定, 208
初期, 205
制限, 811
静的, 206
設定, 204
デフォルト, 205
データベースサーバーメッセージウィンドウでの表示, 215
キャッシュバッファー
パフォーマンス, 204
共通テーブル式
max_recursive_iterations オプション, 645
行番号
[SQL 文] ウィンドウ枠, 843
共有メモリ
CommLinks 接続パラメーター, 327

UNIX 上での保護された接続, 1381
UNIX テンポラリファイル設定, 452
サーバー設定, 289
説明, 97
ターミナルサービス, 98
共用体
クエリエディターでサポートなし, 866
強力な暗号化
AES アルゴリズム, 1397
DatabaseKey (DBKEY) 接続パラメーター, 335
-ec サーバーオプション, 223
-ek データベースオプション, 310
Encryption (ENC) 接続パラメーター, 342
-ep サーバーオプション, 226
Rijndael, 1397
Windows Mobile 上の SQL Anywhere データベース, 1409
アンロードユーティリティ (dbunload), 1044
強力に暗号化されているデータベースの作成, 1398
初期化ユーティリティ (dbinit), 959
説明, 1410
データベースファイル, 1396
強力なテーブル暗号化
初期化ユーティリティ (dbinit), 959
キー配布センター (KDC)
Kerberos 認証での使用, 64
キーペアジェネレーターユーティリティ (createkey)
構文, 981
キーボードショートカット
Interactive SQL, 878
Sybase Central, 823
コードエディター, 825
テキスト補完, 902
キーボードマッピング
説明, 473
キーワード
non_keywords オプション, 650
reserved_keywords オプション, 673
オンにする, 673
使用不可, 650

## く

クイックスタート
トランスポートレイヤーセキュリティ, 1412
バックアップ, 1071
空間データ
st_geometry_asbinary_format オプション, 686
st_geometry_astext_format オプション, 687
st_geometry_asxml_format オプション, 688
st_geometry_describe_type オプション, 689
st_geometry_interpolation オプション, 690
st_geometry_on_invalid オプション, 691
クエリ
Interactive SQL, 848
Interactive SQL から印刷, 872
オプティマイザーバイパス, 657
長時間実行、-zp サーバーオプションを使用した最近のプランの取得, 301
クエリエディター
Transact-SQL のサポートなし, 866
共用体のサポートなし, 866
式エディター, 866
制限事項, 866
説明, 864
クエリ最適化
最近のプランの取得, 301
クエリ内並列処理
-gna オプションによる影響, 238
-gnh オプションによる影響, 239
-gnl オプションによる影響, 240
max_query_tasks オプション, 644
クエリの最適化
オプティマイザーバイパス, 657
クォーラム
データベースミラーリング, 1139
区別
アクセント記号, 496
大文字と小文字, 496
句読表記, 496
組み込みデータベース
起動, 28
接続, 28
データベースサーバー名の指定, 28
クライアント
Kerberos, 63
識別, 321
証明書を信頼するように設定, 1419
トランスポートレイヤーセキュリティを使用する SQL Anywhere の起動, 1422
ミラーリングされたデータベースへの接続, 1163
クライアント API
文字セット変換, 480

クライアント側
DatabaseKey (DBKEY) 接続パラメーター, 335
Encryption (ENC) 接続パラメーター, 342
バックアップ, 1085
バックアップのクイックスタート, 1072
クライアント／サーバー
SQL 文, 479
文字セット変換, 479
クライアント／サーバー通信
言語の問題, 474
クライアントセキュリティ trusted_certificate プロトコルオプション
Mobile Link トランスポートレイヤーセキュリティ, 1430
クライアントでの文のキャッシュ
max_client_statements_cached オプションの設定, 642
説明, 641
クライアントファイル
allow_read_client_file オプション, 590
allow_write_client_file オプション, 592
isql_allow_read_client_file オプション [Interactive SQL], 890
isql_allow_write_client_file オプション [Interactive SQL], 891
read_client_file セキュア機能, 271
READCLIENTFILE 権限, 540
write_client_file セキュア機能, 271
WRITECLIENTFILE 権限, 541
クラスター
説明, 1182
クラスタードハッシュ group by
optimization_workload オプション, 658
クラッシュ
レポート, 1210
グラフィカルなプラン
作成, 867
保存, 867
グループ
PUBLIC, 551
SQL Remote REMOTE パーミッション, 535
SYS, 551
依存性の設定, 1016
オプションの設定, 516
管理, 544
権限, 549
削除, 552
作成, 546
サービス, 178
説明, 512
脱退, 548
追加, 546
テンポラリ領域の制限, 647
統合化ログインの削除, 55
統合化ログインの付与, 51
パスワードを持たない, 550
パーミッション, 549
パーミッションの矛盾, 545
メンバーシップ, 547
メンバーシップの取り消し, 548
ユーザーの追加, 547
ログインポリシーを継承できない, 553
グループ依存性
設定, 1016
グループ作成ウィザード
使用, 546
グローバル証明書
トランスポートレイヤーセキュリティのサーバー証明書として使用, 1418
グローバル署名証明書
トランスポートレイヤーセキュリティ, 1417
グローバルチェックサム
検証, 1122
検証ユーティリティ (dbvalid), 1064
説明, 1120
クワイエットモード
dbisqlc ユーティリティ, 950
Interactive SQL ユーティリティ (dbisql), 976
ping ユーティリティ (dbping), 994
SQL Anywhere スクリプト実行ユーティリティ (dbrunsql), 997
アップグレードユーティリティ (dbupgrad), 1062
アンロードユーティリティ (dbunload), 1044
言語ユーティリティ (dblang), 983
検証ユーティリティ (dbvalid), 1064
サーバーライセンス取得ユーティリティ (dblic), 1003
サーバー列挙ユーティリティ (dblocate), 999
消去ユーティリティ (dberase), 951
情報ユーティリティ (dbinfo), 957
初期化ユーティリティ (dbinit), 959
停止ユーティリティ (dbstop), 1024
データソースユーティリティ (dbdsn), 937
データベースサーバー, 160

トランザクションログユーティリティ (dblog), 1041
バックアップユーティリティ (dbbackup), 913
プロセス生成ユーティリティ (dbspawn), 1023
ログ変換ユーティリティ (dbtran), 985

## け

警告
Mobile Link モニターの警告アイコン, 1242
Relay Server モニターの警告アイコン, 1242
データベースサーバーのパフォーマンス, 1212
モニター, 1301
モニター、電子メールによる通知, 1307
モニター、抑制, 1304
モニターの警告アイコン, 1242
計算カラム
Interactive SQL での新しいローへの追加, 861
Interactive SQL での更新, 862
Interactive SQL での再計算, 862
データベースのアンロード, 1058
データベースのアンロード時の再計算, 1044
継承
権限, 536
パーミッション, 525
ログインポリシーオプション, 555
桁
最大数, 661
結果
データを表示した場合の予期しない記号, 479
結果セット
Interactive SQL でのテーブル値の編集, 859
Interactive SQL でのテーブル編集値の無効化, 861
Interactive SQL のテキスト, 846
印刷, 872
ソート, 864
ローのコピー, 862
ローの削除, 862
ローの挿入, 861
結果を表示した場合の予期しない記号のトラブルシューティング
説明, 479
結合
Interactive SQL での複数の文, 850
権限
BACKUP, 537
DBA, 538
PROFILE, 539
READCLIENTFILE, 540
READFILE, 540
REMOTE DBA, 540
RESOURCE, 541
VALIDATE, 541
WRITECLIENTFILE, 541
管理, 512
継承, 536
説明, 536
取り消し, 532
言語
CHAR 照合のサポート言語の確認, 501
SQL Anywhere のローカライズ版, 468
英語以外の言語用に作成, 503
英語以外のデータベース, 468
大文字と小文字の区別, 480
クライアント／サーバーコンピューティングにおける問題, 474
言語選択ユーティリティ (dblang), 983
指定, 450
ソフトウェアとマニュアル, 469
データベースサーバーの使用言語の確認, 501
データベースの作成, 503
トルコ語, 509
レジストリ設定, 466
ロケール, 485
言語 DLL
レジストリ設定, 983
ロケーション, 461
言語コード
言語ユーティリティ (dblang), 983
言語サポート
説明, 468
マルチバイト文字セット, 487
言語選択ユーティリティ (dblang)
構文, 983
終了コード, 984
言語ラベル
値のリスト, 485
言語リソースライブラリ
メッセージファイル, 474
レジストリ設定, 983
検索
Interactive SQL でのテーブル、カラム、プロシージャー, 857
Sybase Central, 822
データベース, 822

検査制約
説明, 117
データベースのアンロード, 1058
検証
dbvalid ユーティリティ, 1064
Sybase Central からのデータベースの検証, 1118
Sybase Central からのテーブルの検証, 1122
カラム制約, 117
実行のパーミッション, 541
データベース, 1117
トランザクションログ, 1112
バックアップ, 1087
検証ユーティリティ (dbvalid)
構文, 1064
終了コード, 1068
限定ソフトウェア
サポートされる言語, 470

## こ

コア
データベースサーバーが使用するコア数の指定, 245
高可用性
Mobile Link, 1170
NodeType (NODE) 接続パラメーター, 357
SQL Anywhere Veritas Cluster Server エージェント, 1182
Web サービスでの使用, 1205
説明, 1136
データベースミラーリング, 1137
更新
ansi_permissions オプション, 594
ansi_update_constraints オプション, 596
Interactive SQL での値, 860
SQL/2008 の動作, 596
SQL Anywhere の更新のチェック, 907
Transact-SQL パーミッション, 594
更新チェッカー
説明, 907
更新のチェック
説明, 907
高速ランチャー
言語設定の変更, 904
説明, 903
ポート番号の設定, 903
構文
dbbackup ユーティリティ, 913
dbconsole ユーティリティ, 1021
dbdsn ユーティリティ, 936
dberase ユーティリティ, 951
dbfhide ユーティリティ, 953
dbhist ユーティリティ, 955
dbinfo ユーティリティ, 957
dbinit ユーティリティ, 958
dbisqlc ユーティリティ, 949
dbisql ユーティリティ, 975
dblang ユーティリティ, 983
dblic ユーティリティ, 1003
dblocate ユーティリティ, 999
dblog ユーティリティ, 1041
dbns12 ユーティリティ, 921
dbping ユーティリティ, 994
dbrunsql ユーティリティ, 997
dbspawn ユーティリティ, 1022
dbstats ユーティリティ, 991
dbstop ユーティリティ, 1024
dbsupport ユーティリティ, 1027
dbsvc ユーティリティ (Linux), 1007
dbsvc ユーティリティ (Windows), 1013
dbtran ユーティリティ, 985
dbunload ユーティリティ, 1044
dbupgrad ユーティリティ, 1061
dbvalid ユーティリティ, 1064
viewcert ユーティリティ, 934
キーペアジェネレーターユーティリティ (createkey), 981
証明書作成 (createcert), 924
接続パラメーター, 34
接続プロパティのアクセス, 713
データベースサーバープロパティのアクセス, 753
データベースプロパティのアクセス, 781
構文エラー
ジョイン, 623
互換性
ANSI, 584
SQL, 584
Transact-SQL 互換性オプション, 584
互換性オプション
allow_nulls_by_default, 589
ansi_blanks, 592
ansi_close_cursors_on_rollback, 593
ansi_permissions, 594
ansi_substring, 595

ansi_update_constraints, 596
ansinull, 597
ASE 互換性オプション, 585
chained, 603
close_on_endtrans, 605
continue_after_raiserror, 609
conversion_error, 610
divide_by_zero_error, 622
escape_character, 622
fire_triggers, 624
isolation_level, 630
non_keywords, 650
on_tsql_error, 655
quoted_identifier, 668
sql_flagger_error_level, 679
sql_flagger_warning_level, 680
string_rtruncation, 691
time_format, 696
timestamp_format, 698
Transact-SQL 互換性オプション, 585
tsql_outer_joins, 703
tsql_variables, 703
初期設定, 574
データベース互換性オプションのアルファベット順リスト, 585
分類, 575
国際言語サポート
説明, 468
マルチバイト文字セット, 487
国際言語と文字セット
概要, 467
個々のユーザーに割り当てられるログインポリシー
アンロードユーティリティ (dbunload), 1057
固定幅の文字セット
説明, 475
[このコンピューターで稼働しているデータベースに接続]
例, 17
[このコンピューターで稼働しているデータベースに接続]
[接続] ウィンドウ, 14
このコンピューターのデータベースを起動して接続
[接続] ウィンドウ, 14
例, 16
コピー
Interactive SQL でのロー, 862
SQL Anywhere 12 プラグインのデータベースオブジェクト, 831
Windows Mobile デバイスへのデータベースのコピー, 422
テーブルとビュー, 831
コピーノード
NodeType (NODE) 接続パラメーター, 357
-xp オプション, 318
新しい親の割り当て, 1200
親接続の消失, 1200
親の自動割り当て, 1199
親の特定, 1200
監視サーバーとして使用, 1167
削除, 1202
自動的に作成, 1198
手動による追加, 1198
スケールアウトシステムでの消失, 1200
接続, 1201
説明, 1188
チュートリアル, 1190
トランザクションログ処理の状態, 1201
ミラーノードとの違い, 1204
予想を超える子の数, 1201
ログインポリシーの管理, 560
コマンドエコー
echo オプション, 889
コマンドデリミター
dbisqlc ユーティリティ, 950
コマンドデリミター
Interactive SQL ユーティリティ (dbisql), 976
コマンドパラメーターファイル
説明, 909
コマンドファイル
Interactive SQL で開く, 847
コマンドライン
大文字と小文字の区別, 199
設定ファイル内, 201
設定ファイルの使用, 909
データベースサーバー (dbeng12/dbsrv12), 189
データベースサーバーの起動, 199
コマンドラインユーティリティ
Broadcast Repeater (dbns12) の構文, 921
createkey 構文, 981
dbisqlc 構文, 949
Interactive SQL (dbisql) 構文, 975
Linux サービス (dbsvc) の構文, 1007
ping (dbping) の構文, 994

SQL Anywhere コンソール (dbconsole) の構文, 1021
SQL Anywhere スクリプト実行ユーティリティ (dbrunsql) の構文, 997
viewcert 構文, 934
Windows サービス (dbsvc) の構文, 1013
アップグレード (dbupgrad) の構文, 1061
アンロード (dbunload) の構文, 1044
言語選択 (dblang) の構文, 983
検証 (dbvalid) の構文, 1064
サポート (dbsupport) の構文, 1027
サーバー停止 (dbstop) の構文, 1024
サーバーバックグラウンド起動 (dbspawn) の構文, 1022
サーバーライセンス取得 (dblic) の構文, 1003
サーバー列挙 (dblocate) の構文, 999
消去 (dberase) の構文, 951
情報 (dbinfo) の構文, 957
証明書作成ユーティリティ (createcert) の構文, 924
初期化 (dbinit) の構文, 958
データソース (dbdsn) の構文, 936
トランザクションログ (dblog) の構文, 1041
バックアップ (dbbackup) の構文, 913
パフォーマンス統計値 (dbstats) の構文, 991
ヒストグラム (dbhist) の構文, 955
ファイル非表示 (dbfhide) の構文, 953
ログ変換 (dbtran) の構文, 985
コミット
COOPERATIVE_COMMIT_TIMEOUT オプション, 610
cooperative_commits オプション, 611
delayed_commit_timeout オプション, 619
delayed_commits オプション, 620
コミットされていないトランザクションリカバリ, 1090
コメント
Interactive SQL での追加と削除, 849
監査, 1392
コンソールユーティリティ (dbconsole)
構文, 1021
使用, 905
設定, 906
コードエディター
外観のカスタマイズ, 825
キーボードショートカット, 825
説明, 824
開く, 824
フォント, 825
コード化
PKI オブジェクト, 934
文字セット, 473
コードページ
ANSI, 476
default_isql_encoding オプション, 888
Interactive SQL ユーティリティ (dbisql), 976
OEM, 476
UNIX プラットフォームでの推奨, 508
Windows, 476
Windows プラットフォームでの推奨, 507
概要, 474
説明, 473
コールバック関数
データベースサーバーオプション (dbeng12、dbsrv12), 229

## さ

再帰クエリ
max_recursive_iterations オプション, 645
再構築
Windows Mobile のデータベース, 423
認証データベース, 186
再実行ログ
説明, 127
最小要件
ハードウェア, 814
最大
データベースサイズ, 811
データベースファイルサイズ, 811
最適化
最近のプランの取得, 301
再表示
Sybase Central, 177
間隔の設定, 177
自動, 177
再呼び出し
Interactive SQL の文, 852
サイレント
データベースサーバー, 160
削除
ANSI の動作, 594
DB 領域, 126
Interactive SQL でのローの使用, 862
Kerberos ログイン, 68
Linux サービス, 1007

Transact-SQL パーミッション, 594
グループ, 552
サービス, 1013
消去ユーティリティ (dberase), 951
データベース, 140
データベースファイル, 140
テーブルのロー, 862
統合化ログイン, 55
ユーザー, 517
作成
dbdsn を使用して ODBC データソースを作成, 936
dbinit を使用したデータベースの作成, 958
DB 領域, 123
Kerberos ログイン, 67
ODBC データソースアドミニストレーターを使用した作成, 39
SQL のデータベース, 118
Windows Mobile 用データベース, 418
新しい証明書, 924
既存のデータベースを使用した強力に暗号化されているデータベース, 1398
既存のデータベースを使用した、テーブルが暗号化されているデータベース, 1406
強力に暗号化されているデータベース, 1398
グループ, 546
コマンドラインからのデータベース, 119
[接続] ウィンドウを使用した ODBC データソース, 38
接続プロファイル, 10
データベースのチュートリアル, 108
ユーザー, 513
ログインポリシー, 557
サブフォルダー
Windows Mobile, 459
サポートされる機能
Windows Mobile, 434
サポートされるプラットフォーム
Kerberos, 63
SQL Anywhere SNMP Extension Agent, 1319
サポート対象外の機能
Windows Mobile での SQL Anywhere の制限, 435
サポートユーティリティ (dbsupport)
構文, 1027
使用, 1210
サンプル
Windows の [スタート] メニューからアクセス, 459
環境変数, 454
ソース指定, 444
ロケーション, 458
サンプルアプリケーション
ADO.NET Sample, 408
ESQL Sample, 409
ODBC Sample, 410
Windows Mobile, 406
サンプルディレクトリ
環境変数, 454
説明, 459
サンプルデータベース
demo.db の起動, 1
Mac OS X での接続, 9
UNIX での接続, 7
Windows Mobile の 2 つのバージョン, 406
チュートリアル, 1
サーバー
管理, 1013
起動時のトラブルシューティング, 1211
検索, 999
自動的に起動, 80
接続しないでデータベースを起動, 148
接続の制限, 237
代替サーバー名の指定, 315
停止するために必要なパーミッション, 235
データベース機能の無効化, 1387
データベースの停止, 149
データベースミラーリングによる高可用性, 1137
トランスポートレイヤーセキュリティを使用する起動, 1420
名前の制限, 257
名前のトランケーションの長さ, 258
バッチファイルから起動, 1022
プロパティ, 753
読み込み専用アクセス、データベースミラーリング, 314
サーバーアドレス
DSEdit, 1440
サーバーオプション
Windows Mobile でサポート対象外のオプション, 437
Windows Mobile データベースに対する指定, 1408
データベース, 305

データベースサーバー (dbeng12/dbsrv12), 189
リカバリ, 228
サーバー側
-ec サーバーオプション, 223
-ek サーバーオプション, 310
-ep サーバーオプション, 226
-es サーバーオプション, 227
バックアップ, 1079
バックアップのクイックスタート, 1071
並列バックアップ, 1116
[サーバー起動オプション] ウィンドウ
Linux での使用, 286
Windows Mobile での使用, 427
サーバー情報
sasrv.ini, 91
サーバー証明書
トランスポートレイヤーセキュリティにおけるグローバル証明書の使用, 1418
サーバー停止ユーティリティ (dbstop)
SQLCONNECT の使用, 1026
構文, 1024
終了コード, 1026
使用, 146
パーミッション, 235
サーバー統計
SQL Anywhere SNMP Extension Agent による取得, 1328
SQL Anywhere SNMP Extension Agent の OID, 1337
サーバー認証
Mobile Link トランスポートレイヤーセキュリティ, 1429
SQL Anywhere トランスポートレイヤーセキュリティ, 1422
サーバーの確認
SQL Anywhere トランスポートレイヤーセキュリティ, 1422
サーバーのマルチプログラミングレベル
自動チューニング, 165
サーバーのモニタリング
Mobile Link、説明, 1217
Relay Server、説明, 1217
説明, 1217
サーバーのロングネーム
dblocate での表示, 999
サーバーバックグラウンド起動ユーティリティ (dbspawn)
構文, 1022
終了コード, 1024
サーバープロパティ
SQL Anywhere SNMP Extension Agent による取得, 1328
SQL Anywhere SNMP Extension Agent による設定, 1329
SQL Anywhere SNMP Extension Agent の OID, 1341
アルファベット順リスト, 753
大文字と小文字の区別, 753
レポート, 994
サーバー名
-n オプション, 257
大文字と小文字の区別, 152
サーバー列挙ユーティリティ (dblocate), 999
停止 (dbstop) の構文, 1024
サーバーメッセージ
Linux-サーバーメッセージウィンドウの表示, 286
Linux-サーバーメッセージウィンドウの表示とデータベースの起動, 284
Mac OS X での表示, 284
Solaris での表示, 282
キャッシュウォーミング, 216
シェルモードでの Linux の実行, 282
表示, 266
ファイルへの出力, 258
ファイルへの出力とファイルのトランケート, 262
ロギング起動エラー, 259
ログファイルサイズの制限, 261
ログファイルの名前変更と再開, 260
[サーバーメッセージと実行された SQL] ウィンドウ枠
説明, 154
サーバーライセンス取得ユーティリティ (dblic)
構文, 1003
終了コード, 1006
サーバーライセンスユーティリティ (dblic)
Vista での権限の昇格の必要性, 188
Windows 7 での権限の昇格の必要性, 188
Windows Server 2008 での権限の昇格の必要性, 188
サーバー列挙ユーティリティ (dblocate)
構文, 999
終了コード, 1002
サービス
Linux サービスの依存性の設定, 1008

Linux サービスの起動, 1007
Linux サービスのリスト, 1007
Linux でのサービスタイプの設定, 1008
PID ファイル, 1011
Sybase Central からの作成, 172
Vista での実行, 188
Windows, 171
Windows 7 での実行, 188
Windows Server 2008 での実行, 188
Windows サービスマネージャーの使用, 178
Windows でのサービスタイプの設定, 1016
Windows への追加, 172
アカウント, 175
新しいデータベースの追加, 176
依存性, 179
依存性の設定, 1016
イベントログ, 155
オプション, 175
開始, 177
起動オプション, 174
起動順序, 179
起動障害, 175
グループ, 178
グループ依存性の設定, 1016
再表示, 177
削除, 173
サービスユーティリティ (dbsvc), 1013
実行ファイル, 176
セキュリティ, 175
設定, 174
説明, 170
ターミナルサービス, 98
停止, 177
適格なプログラム, 171
デスクトップのアイコン, 175
データベースサーバー, 170
パラメーター, 174
複数, 179
リスト, 1013
レジストリ設定, 464
サービスグループ
説明, 178
サービス作成ウィザード
Windows Mobile でサポート対象外, 440
使用, 172
サービスとしてログインする権限
Linux サービスユーティリティ (dbsvc), 1008
Windows サービスユーティリティ (dbsvc), 1016
サービスユーティリティ (dbsvc)
Linux オプション, 1007
Linux の構文, 1007
Vista での権限の昇格の必要性, 188
Windows 7 での権限の昇格の必要性, 188
Windows Server 2008 での権限の昇格の必要性, 188
Windows オプション, 1013
Windows 構文, 1013
終了コード, 1020

## し

シェルモード
データベースサーバーメッセージの表示, 282
式エディター
クエリエディター, 866
識別
クライアントアプリケーション, 321
識別子
大文字と小文字を区別しない, 480
国際的な側面, 480
シグネチャー
認証アプリケーションの取得, 182
自己署名証明書
トランスポートレイヤーセキュリティ, 1414
トランスポートレイヤーセキュリティ用に作成, 1415
自己署名証明書の設定
トランスポートレイヤーセキュリティ, 1415
システムイベント
BackupEnd, 1127
Connect, 1127
ConnectFailed, 1127
DatabaseStart, 1127
DBDiskSpace, 1127
Deadlock, 1127
Disconnect, 1127
GlobalAutoincrement, 1128
GrowDB, 1128
GrowLog, 1128
GrowTemp, 1128
LogDiskSpace, 1127
MirrorFailover, 1128
MirrorServerDisconnect, 1128
RAISERROR, 1128

ServerIdle, 1128
TempDiskSpace, 1127
説明, 1126
定義, 1123
データベースミラーリング, 1173
内部, 1132
システムオブジェクト
アンロード, 1057
システム障害
説明, 1106
リカバリ, 1106
システム情報ファイル
DSN 接続パラメーターでの指定, 338
Mac OS X でのデータソースの作成, 45
暗号化されたパスワードの保管, 341
説明, 48
データソースユーティリティ (dbdsn) の使用, 940
パスワードの保管, 360
システムテーブル
preserve_source_format, 663
prevent_article_pkey_update, 664
ソースカラム, 663
システムの稼働条件
Veritas Cluster Server エージェント, 1183
システムビュー
パーミッション, 518
ユーザーとグループ, 518
事前定義の DB 領域
説明, 120
実行
Interactive SQL での複数の文, 850
Interactive SQL での文, 848
Interactive SQL の文, 848
イベントハンドラー, 1133
コマンドファイル, 851
実行時間
isql_command_timing オプション, 892
実行スレッド
-gna dbsrv12 サーバーオプション, 238
-gn dbsrv12 サーバーオプション, 237
-gnh dbsrv12 サーバーオプション, 239
-gnl dbsrv12 サーバーオプション, 240
について, 161
[実行] ツールバーボタンの設定
説明, 849
実行プラン
印刷, 872
実行プログラム
Vista 用に署名, 188
Windows 7 用に署名, 188
Windows Server 2008 用に署名, 188
設定ファイルでの条件付き解析の使用, 910
設定ファイルの使用, 909
ロケーション, 458
実行プログラム名
サーバーライセンス取得ユーティリティ (dblic), 1003
失敗ログイン
DBA, 538
質問
文字セット, 477
自動化
管理タスク, 1123
自動リカバリ
説明, 1071
シナリオ
データベースミラーリングシステムでのフェールオーバー, 1175, 1178
シャープ記号
設定ファイルでの使用, 909
修飾された名前
データベースオブジェクト, 562
テーブル, 549
重大度
モニター, 1242
修復
モニターのリソース, 1260
終了コード
Interactive SQL ユーティリティ (dbisql), 981
ping ユーティリティ (dbping), 997
Windows サービスユーティリティ (dbsvc), 1020
アップグレードユーティリティ (dbupgrad), 1064
アンロードユーティリティ (dbunload), 1058
言語ユーティリティ (dblang), 984
検証ユーティリティ (dbvalid), 1068
サーバー停止ユーティリティ (dbstop), 1026
サーバーバックグラウンド起動ユーティリティ (dbspawn), 1024
サーバーライセンス取得ユーティリティ (dblic), 1006
サーバー列挙ユーティリティ (dblocate), 1002
消去ユーティリティ (dberase), 953
情報ユーティリティ (dbinfo), 958

初期化ユーティリティ (dbinit), 975
データソースユーティリティ (dbdsn), 940
トランザクションログユーティリティ (dblog), 1044
バックアップユーティリティ (dbbackup), 920
パフォーマンス統計値ユーティリティ (dbstats), 993
ヒストグラムユーティリティ (dbhist), 956
ログ変換ユーティリティ (dbtran), 991
手動イベント
作成, 1134
取得
Interactive SQL の文, 852
準備文
max_statement_count オプション, 646
サーバーでサポートされる最大数, 811
接続制限, 552
障害
リカバリ, 1071
昇格操作エージェント
Vista, 188
Windows 7, 188
Windows Server 2008, 188
消去
Windows Mobile デバイスのデータベース, 426
データベース, 140
消去ユーティリティ (dberase)
構文, 951
終了コード, 953
使用, 141
条件付き解析
設定ファイル, 910
照合
Adaptive Server Enterprise の照合, 493
CHAR 照合のかくにん, 501
Mac OS X の推奨, 506
NCHAR 照合のかくにん, 501
SQL Anywhere データベース, 491
UNIX プラットフォームでの推奨, 508
Windows プラットフォームでの推奨, 507
サポートされている照合のリスト, 492
サポートされているトルコ語照合の違い, 511
初期化中に適合化, 958
説明, 473
選択, 496
代替, 492
デフォルト, 135
デフォルトの判断, 501
データベースの作成, 503
ドイツ語データベースのデフォルト, 487
トルコ語データベース, 509
変更, 504
マルチバイト, 487
文字のソート, 487
照合順
初期化ユーティリティ (dbinit), 959
照合の適合化
ICU, 489
照合の選択
考慮事項, 496
説明, 496
照合の適合化
dbinit ユーティリティ, 958
Windows Mobile でのサポートの制限, 416
オプション, 496
カタカナ, 496
スウェーデン語, 496
説明, 496
日本語, 496
ひらがな, 496
照合の変更
説明, 504
冗長モード
アンロードユーティリティ (dbunload), 1044
使用法
表示, 203
上方コードページ
説明, 474
情報ユーティリティ (dbinfo)
構文, 957
終了コード, 958
証明書
certificate_company プロトコルオプション, 374
certificate_name プロトコルオプション, 375
certificate_unit プロトコルオプション, 376
ECC 証明書の作成, 924
RSA 証明書の作成, 924
trusted_certificate プロトコルオプション, 401
X.509 の表示, 934
トランスポートレイヤーセキュリティのデジタル証明書, 1414
証明書作成ユーティリティ (createcert)
構文, 924
証明書失効リスト
表示, 934
証明書チェーン

トランスポートレイヤーセキュリティ, 1415
証明書ビューアーユーティリティ (viewcert)
構文, 934
証明書フィールドの確認
Mobile Link トランスポートレイヤーセキュリティ, 1430
SQL Anywhere トランスポートレイヤーセキュリティ, 1422
証明書ユーティリティ
トランスポートレイヤーセキュリティ, 1434
証明書要求
表示, 934
証明書を信頼するようにクライアントを設定する
トランスポートレイヤーセキュリティ, 1419
初期化
データベース, 113
初期化ユーティリティ (dbinit)
Windows Mobile 用データベースの作成, 421
構文, 958
終了コード, 975
使用, 119
署名
ECC 証明書と RSA 証明書, 924
Vista 用実行プログラム, 188
Windows 7 用実行プログラム, 188
Windows Server 2008 用実行プログラム, 188
署名付き証明書
トランスポートレイヤーセキュリティでの作成, 1417
署名付き証明書の作成
トランスポートレイヤーセキュリティ, 1417
所有者
説明, 525
ショートカット
Interactive SQL, 878
Sybase Central, 823
シングルステップ
説明, 849
シングルバイト文字セット
説明, 474
進行メッセージ
設定, 665
バックアップ用に有効化, 913
ユーティリティデータベース, 666
診断ディレクトリ
SADIAGDIR 環境変数, 449
シーケンスジェネレーター
パーミッション, 528

## す

スイッチ
Broadcast Repeater ユーティリティ (dbns12), 921
dbisqlc ユーティリティ, 950
Interactive SQL ユーティリティ (dbisql), 976
Linux サービスユーティリティ (dbsvc), 1007
ping ユーティリティ (dbping), 992
SQL Anywhere コンソールユーティリティ (dbconsole), 1021
SQL Anywhere スクリプト実行ユーティリティ (dbrunsql), 997
Windows サービスユーティリティ (dbsvc), 1013
アップグレードユーティリティ (dbupgrad), 1062
アンロードユーティリティ (dbunload), 1044
言語選択ユーティリティ (dblang), 983
検証ユーティリティ (dbvalid), 1064
サポートユーティリティ (dbsupport), 1027
サーバー停止ユーティリティ (dbstop), 1024
サーバーバックグラウンド起動ユーティリティ (dbspawn), 1023
サーバーライセンス取得ユーティリティ (dblic), 1003
サーバー列挙ユーティリティ (dblocate), 999
消去ユーティリティ (dberase), 951
情報ユーティリティ (dbinfo), 957
証明書作成 (createcert), 924
証明書ビューアーユーティリティ (viewcert), 934
初期化ユーティリティ (dbinit), 959
データソースユーティリティ (dbdsn), 936
トランザクションログユーティリティ (dblog), 1041
バックアップユーティリティ (dbbackup), 913
ヒストグラムユーティリティ (dbhist), 955
ログ変換ユーティリティ (dbtran), 985
推定
user_estimates オプション, 706
スウェーデン語
UCA 照合, 496
数値の精度
データベースオプション, 661
スキーマ

定義のアンロード, 1044
比較, 832
比較して同じにする, 833
スクリプト実行ユーティリティ (dbrunsql)
構文, 997
スケジュールされたイベント
スケジュール作成ウィザード, 1126
スケジュール
イベント, 1123
イベントスケジュールの定義, 1126
サーバーの停止, 280
スケジュール作成ウィザード, 1126
説明, 1124
定義, 1123
内部, 1132
バックアップ, 1103
スケジュールとイベントの使用によるタスクの自動化
説明, 1123
スケジュール作成ウィザード
使用, 1126
スケジュールされたイベント
説明, 1124
データベースミラーリング, 1140
夏時間, 1133
スケジュールされていないタスク
説明, 165
スケジュールされていない要求
説明, 165
スケールアウト
NodeType (NODE) 接続パラメーター, 357
SQL Anywhere, 1188
Web サービスでの使用, 1205
-xp オプション, 318
親接続の消失, 1200
親の自動割り当て, 1199
コピーノード, 1188
コピーノードの親の特定, 1200
サーバーの変更, 1202
接続プールでの使用, 75
チュートリアル, 1190
データベースミラーリングでの使用, 1203
ログインポリシーの管理, 560
スタックオーバーフロー
エラー, 243
スタックサイズ
外部関数, 234
最大, 243
ステップスルー
Interactive SQL, 848
ステータス
メンテナンスプラン, 1104
ステータス情報ファイル
データベースミラーリング, 1145
プライマリサーバーを決定するときのロール, 1164
ストアドプロシージャー
SQL Anywhere SNMP Extension Agent による実行, 1330
セキュリティ機能, 1379
パーミッションの設定, 527
ストアドプロシージャー
データベースドキュメントの生成, 838
ストレージエリアネットワーク
データベースファイルの格納, 127
ストレージカード
Windows Mobile, 404
スナップショット
ポイントインタイム, 1084
スナップショットアイソレーション
isolation_level データベースオプション, 630
updatable_statement_isolation オプション, 704
スペース
接続文字列, 34
スマートフォン
SQL Anywhere サーバーの制限, 406
スレッド
-gna dbsrv12 サーバーオプション, 238
-gn dbsrv12 サーバーオプション, 237
-gnh dbsrv12 サーバーオプション, 239
-gnl dbsrv12 サーバーオプション, 240
Linux, 163
SQL Anywhere でのスレッド, 162
UNIX の動作, 163
Windows, 163
動作の制御, 164
複数のプロセッサー, 244
スレッドアプリケーション
UNIX 用 dbping_r, 994
スレッド化
Linux, 163
UNIX の動作, 163
Windows, 163
説明, 161
動作の制御, 164
スレッドデッドロック

マルチプログラミングレベルとの関係, 169
スレッド動作の制御
説明, 164

## せ

制限
SQL Anywhere, 811
Windows Mobile 5.0 for Smartphone 上の SQL Anywhere, 406
Windows Mobile での SQL Anywhere の実行, 435
暗号化キー, 1402
識別子, 811
テンポラリファイル, 695
データベースミラーリング, 1140
パスワード, 515
バックアップ中, 1079
メモリ, 814
リカバリ中, 1079
制限事項
SQL Anywhere, 811
モニター, 1220
正常性と統計情報
表示, 837
モニター, 1217
生成
ECC 証明書, 924
Interactive SQL での SQL 文, 859
RSA 証明書, 924
制約
カラムとテーブル, 117
制約の選択
説明, 117
セキュリティ
AES 暗号化, 1397
DatabaseKey (DBKEY) 接続パラメーター, 335
-ec サーバーオプション, 223
-ek サーバーオプション, 310
Encryption (ENC) 接続パラメーター, 342
-ep サーバーオプション, 226
-es サーバーオプション, 227
FIPS, 1412
Vista での SQL Anywhere の実行, 187
Windows 7 での SQL Anywhere の実行, 187
Windows Mobile, 1408
Windows Server 2008 での SQL Anywhere の実行, 187
イベントの例, 1129
概要, 1379
監査, 1388
監査オプション, 598
監査の検索, 1390
サービス, 175
システム関数, 1380
失敗 DBA ログイン, 538
説明, 1379
単純暗号化の設定ファイルへの追加, 953
テンポラリファイル, 451
データのアンロード, 1396
データのロード, 1396
データベース機能の無効化, 1387
データベースサーバー, 1395
データベースの削除, 1395
データベースの作成, 1395
データベースファイルの暗号化, 1396
データベースファイルのコピー, 74
統合化ログイン, 1382
トランスポートレイヤーセキュリティの説明, 1410
パスワード, 1383
パスワードの最小長, 648
ビュー, 564
ヒント, 1380
ファイルアクセス, 236
ファイル非表示ユーティリティ (dbfhide), 953
プロシージャー, 564
ユーティリティデータベース, 561
設計
データベース, 113
データベース、考慮事項, 113
設計モード
説明, 830
接続
BroadcastListener (BLISTENER) プロトコルオプション, 373
ClientPort (CPORT) プロトコルオプション, 377
dblocate と LDAP, 1002
dblocate を使用したトラブルシューティング, 999
dbping を使用したトラブルシューティング, 994
dedicated_task オプション, 617
Embedded SQL のパフォーマンスのテスト, 93
Host (IP) プロトコルオプション, 381
Interactive SQL からの接続, 14

IPv6 アドレス, 100
LDAP の使用, 103
LDAP プロトコルオプション, 387
login_procedure オプションを使用した最大数の設定, 637
RAS, 102
ServerPort (PORT) プロトコルオプション, 396
SQL Anywhere OEM Edition 用に認証, 184
SQL Anywhere コンソールユーティリティから, 14
Sybase Central からの接続, 14
Sybase Central 接続プロファイル, 11
TCP/IP, 24
-ti サーバーオプションを使用して切断, 277
Windows Mobile, 411
Windows Mobile デバイス, 411
Windows Mobile データベースとデスクトップアプリケーション, 36
Windows Mobile と ODBC データソース, 47
概要, 5
活性, 278
監査, 1390
機能の保護, 1380
組み込みデータベース, 28
コピーノード, 1201
制限, 237
[接続] ウィンドウの概要, 14
接続しないでデータベースを起動, 148
説明, 5
定義, 5
テナントデータベースへ管理ツールを介し, 24
デフォルトパラメーター, 33
テンポラリ, 76
テンポラリファイル最大領域, 695
テンポラリファイル領域の制限, 695
テンポラリ領域の制限, 647
データソースの使用, 18
データベース, 5
データベース機能の有効化, 677
データベースサーバーの起動例, 145
データベースサーバーの検出, 80
データベースサーバーを自動的に起動, 80
データベース接続シナリオ, 30
データベースの自動的な起動, 28
データベースミラーリング, 1163
統合化ログイン, 1382
トラブルシューティング, 76
認証, 608
パフォーマンス, 91
必要な接続パラメーター, 32
ファイアウォール, 101
プログラミングインターフェイス, 49
プロパティ, 713
プロパティのアルファベット順リスト, 713
プール, 74
命名, 332
文字セット, 478
問題点, 76
ユーティリティからの接続, 29
ユーティリティデータベース, 137
読み込み専用のスケールアウト, 1201
ログインポリシー, 553
ローカルサーバーの起動, 16, 17
ローカルデータベース, 5, 16, 17
接続 ID
説明, 30
[接続] ウィンドウ
概要, 14
開く, 14
接続しているユーザー
管理, 542
接続しているユーザーの管理
説明, 542
接続シナリオ
概要, 30
接続の削除
データベース, 543
モニター, 1247
接続の失敗
トラブルシューティング, 76
接続の切断
データベース, 543
別のユーザーとデータベースの接続, 543
モニター, 1247
接続パラメーター
dbisqlc ユーティリティ, 950
Delphi, 942
DescribeCursor, 942
Driver, 942
Escape, 942
GetTypeInfoChar, 942
InitString, 942
Interactive SQL ユーティリティ (dbisql), 976
IsolationLevel, 942
KeysInSQLStatistics, 942
LazyAutocommit, 942

ODBC データソース, 942
ping ユーティリティ (dbping), 994
PrefetchOnOpen, 942
PreventNotCapable, 942
SQL Anywhere, 320
SQL Anywhere コンソールユーティリティ (dbconsole), 1021
SuppressWarnings, 942
TranslationDLL, 942
TranslationName, 942
TranslationOption, 942
アップグレードユーティリティ (dbupgrad), 1062
アルファベット順リスト, 320
アンロードユーティリティ (dbunload), 1044
大文字と小文字の区別, 320
大文字と小文字の区別なし, 321
概要, 30
空の値, 35
検証ユーティリティ (dbvalid), 1064
情報ユーティリティ (dbinfo), 957
接続の確立, 49
接続文字列, 31
設定, 31
説明, 320, 942
停止ユーティリティ (dbstop), 1024
デフォルトパラメーターの使用, 33
データソース, 36
データソースユーティリティ (dbdsn), 936
バックアップユーティリティ (dbbackup), 913
ブール値, 321
優先度, 35
ルール, 34
ログ変換ユーティリティ (dbtran), 985
ロケーション, 78
接続パラメーター
構文, 34
シナリオ, 32
接続プロパティ
アルファベット順リスト, 713
大文字と小文字の区別, 713
レポート, 994
接続プロファイル
インポート, 13
エクスポート, 14
作成, 10
自動的に接続, 11
説明, 9
編集, 12
接続プール
ADO.NET Sample の使用, 408
ConnectionPool (CPOOL) 接続パラメーター, 332
SQL Anywhere, 74
Windows Mobile でサポート対象外, 435
読み込み専用のスケールアウトでの使用, 75
接続文字列
空の接続パラメーター, 35
スペース, 34
接続パラメーターのアルファベット順リスト, 320
接続パラメーターの使用, 30
説明, 31
重複するパラメーターの優先順位, 35
表現, 31
文字セット, 478
接続文字列を使用して接続
[接続] ウィンドウ, 14
切断
SQL Anywhere コンソールユーティリティからユーザーを切断, 543
データベースの停止, 149
切断された接続
SQL Anywhere SNMP Extension Agent トラップ, 1331
設定
interfaces ファイル, 1438
ODBC データソース, 39
sql.ini, 1438
SQL Anywhere コンソールユーティリティ (dbconsole), 906
Windows Mobile のデータベース, 416
再表示間隔, 177
テキスト補完, 902
テンポラリオプション, 570
データベースオプション, 568
設定ファイル
dbfhide を使用した単純暗号化の追加, 953
エスケープ文字, 910
オプション, 201
条件付き解析, 910
説明, 909
非表示, 953
セットのキーワード
オフにできない, 650
接続パラメーター

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

SQL Anywhere スクリプト実行ユーティリティ (dbrunsql), 997
セミコロン
接続文字列での使用, 31
セルのコピー
Interactive SQL, 862
選択したローのコピー
Interactive SQL, 862
選択性推定
user_estimates オプション, 706

## そ

送信
iAnywhere へのエラーレポート, 1210
モニター、警告の電子メール, 1307
挿入
Interactive SQL でのテーブルへのローの挿入, 861
速度プロパティ
説明, 810
組織
certificate_company プロトコルオプション, 374
組織単位
certificate_unit プロトコルオプション, 376
Mobile Link トランスポートレイヤーセキュリティでの確認, 1430
ソフトウェア
更新, 907
データベースサーバーのライセンス, 1003
バージョン, 287
ソフトウェア更新のチェック
説明, 907
ソフトウェアの更新
入手, 907
ソフトウェアのバージョン
判別, 1068
ソフトウェアバージョン
データベースサーバー, 287
ソフトウェアライセンス
サーバーのライセンス, 1003
ソース制御
Interactive SQL からのファイルのチェックアウト, 876
Interactive SQL からのファイルのチェックイン, 877
Interactive SQL で設定, 874
Interactive SQL と統合, 873
プロジェクトの使用, 875
ソース制御プロジェクト
Interactive SQL から利用可能なアクション, 877
ソート
結果セット, 864
ソート順
照合, 468
説明, 473

## た

第 1 ロー最適化オプション
optimization_goal, 655
ダイアルアップネットワーキング
接続, 102
代替サーバー名
-sn オプション, 315
タイムアウト
request_timeout オプション, 672
トラブルシューティング, 1215
タイムスタンプ
データベースファイル, 127
トランザクションログ, 127
タイムゾーン
SQL Remote の TIMESTAMP WITH TIME ZONE 表示フォーマット, 684
TIMESTAMP WITH TIME ZONE 表示フォーマット, 700
調整, 697
楕円曲線暗号化の証明書
作成, 924
表示, 934
高い可用性
ライブバックアップ, 1075
タスク
SQL Anywhere でのスレッド, 161
イベント, 1133
スケジュール, 1133
タスクリスト
表示, 820
ダッシュボード
モニター, 1243
脱退
グループ, 548
単一オプションキーワード
オフにできない, 650
探索条件
user_estimates オプション, 706

単純暗号化
Windows Mobile 上の SQL Anywhere データベース, 1409
説明, 1396
断片化
パフォーマンス, 125
ダーティページ
説明, 133
ターミナルサービス
共有メモリ接続, 98

## ち

チェックアウト
Interactive SQL からのファイルのチェックアウト, 876
チェックイン
Interactive SQL からのファイルのチェックイン, 877
チェックサム
-wc サーバーオプション, 288
-wc データベースオプション, 317
Windows Mobile データベースで自動的に有効化, 1121
Windows Mobile データベース用に有効化, 418
検証ユーティリティ (dbvalid), 1064
自動, 1121
初期化ユーティリティ (dbinit), 959
説明, 1120
チェックサムの検証
説明, 1122
チェックポイント
checkpoint_time オプション, 603
間隔, 232
緊急度, 1114
説明, 132
トランザクションログの削除, 255
バックアップ, 1113
バックアップ中は許可されない, 1079
パーソナルサーバーとネットワークサーバーの違い, 142
チェックポイント専用モード
データベースサーバー, 247
チェックポイントログ
説明, 133
違い
ミラーノードとミラーノード, 1204
置換文字
on_charset_conversion_failure オプション, 654
チケット付与権限付きチケット
Kerberos, 65
致命的なエラー
-uf サーバーオプション, 283
レポート, 1210
抽出ユーティリティ (dbxtract)
Windows Mobile でサポート対象外, 441
チュートリアル
Interactive SQL を使用した Windows Mobile データベースの管理, 432
Sybase Central からの Windows Mobile データベースの実行, 427
サンプルデータベースへの接続, 1
データベースの作成, 108
データベースミラーリング, 1146
複数データベースのデータベースミラーリング, 1154
モニター, 1222
読み込み専用スケールアウトシステム, 1190
チューニング
マルチプログラミングレベル (手動), 169
マルチプログラミングレベル (動的), 166
長時間実行クエリ
-zp サーバーオプションを使用した最近のプランの取得, 301

## つ

追加
ECC 証明書と RSA 証明書, 924
Interactive SQL での新しいロー, 861
通称
certificate_name プロトコルオプション, 375
Mobile Link トランスポートレイヤーセキュリティでの確認, 1430
通信
DatabaseKey (DBKEY) 接続パラメーター, 335
-ec サーバーオプション, 223
Encryption (ENC) 接続パラメーター, 342
サポート, 96
説明, 96
デバッグ, 296
データベースサーバー, 289
トラブルシューティング, 1215
プロトコルオプション, 369
通信の圧縮
Compress (COMP) 接続パラメーター, 329

CompressionThreshold (COMPTH) 接続パラメーター, 330
ツールバー
Interactive SQL の [実行] ボタン, 849

## て

停止
Windows Mobile 上のサーバー, 431
時刻の指定, 280
データベース, 1024
データベース (ASTOP), 324
ディスク
障害からのリカバリ, 1089
断片化とパフォーマンス, 125
ディスクキャッシュ
オペレーティングシステム, 281
ディスクコントローラ
トランザクションログの管理, 129
ディスクのクラッシュ
説明, 1106
ディスクフル
コールバック関数, 229
トランザクションログへの書き込みエラー, 128
ディスクミラーリング
トランザクションログ, 129
ディスク領域
イベントの例, 1129
ディスク上のテーブルのサイズの判断に dbinfo を使用, 958
ファイルシステムフルのコールバック関数, 229
ディレクトリアクセスサーバー
Windows Mobile でサポート対象外, 435
ディレクトリ構造
SQL Anywhere, 458
テキストファイルフォーマット
Interactive SQL 出力, 898
Interactive SQL 入力, 890
テキストプラン
プランビューアーを使用した Ultra Light テキストプラン, 867
テキスト補完
キーボードショートカット, 902
使用, 901
設定, 902
テクニカルサポート
テクニカルサポートのためのデータベースの暗号化, 1400
データベースの復号化, 1401
デジタル証明書
トランスポートレイヤーセキュリティ, 1414
デジタル証明書の作成
トランスポートレイヤーセキュリティ, 1414
デジタル署名
Mobile Link トランスポートレイヤーセキュリティ, 1429
SQL Anywhere トランスポートレイヤーセキュリティ, 1422
デッドロック
Deadlock システムイベント, 1127
log_deadlocks オプション, 634
デッドロックレポート
log_deadlocks オプション, 634
テナントデータベース
接続, 24
デバッグ
debug_messages オプション, 616
Interactive SQL での SQL 文, 848
SQL スクリプト, 975
Web サービスクライアント, 300
イベントハンドラー, 1131
デバッグモード
説明, 830
デフォルト
接続パラメーター, 33
ポート, 396
デフォルトのデータベースサーバー
説明, 257
デフォルトの文字セット
UNIX, 487
Windows, 487
説明, 487
デリミター
文, 887
電子メール
モニターのユーザー, 1299
電子メール送信
モニター、警告の通知, 1307
転送ログ
説明, 127
テンポラリオプション
Kerberos ログインセキュリティ, 73
スコープと継続期間, 570
設定, 568

統合化ログインセキュリティ, 73
テンポラリテーブル
制限, 811
説明, 76
テンポラリファイル
-dt サーバーオプションによるロケーションの指定, 221
SATMP 環境変数を使用したロケーションの指定, 451
temp_space_limit_check チェックオプション, 695
TEMP 環境変数を使用したロケーションの指定, 457
TMPDIR 環境変数を使用したロケーションの指定, 457
TMP 環境変数を使用したロケーションの指定, 457
UNIX 共有メモリ接続, 452
Windows Mobile でのロケーション, 466
制限, 811
セキュリティ, 451
接続で使用される最大領域, 695
テンポラリ領域
制限, 647
データ型
制限, 811
選択, 115
データ型変換
エラー, 610
データ整合性
カラム制約, 117
データソース
dbdsn を使用して ODBC データソースを作成, 936
Embedded SQL, 37
Interactive SQL ユーティリティ (dbisql), 976
Mac OS X, 45
ODBC, 36
ODBC 接続パラメーター, 942
ODBC データソースアドミニストレーターを使用した生成, 39
SQL Anywhere コンソールユーティリティ (dbconsole), 1021
UNIX, 48
Windows Mobile での使用, 47
Windows Mobile 用に作成, 412
[接続] ウィンドウを使用した生成, 38
接続パラメーター, 30
説明, 36
データソースユーティリティ (dbdsn), 936
ファイル, 45
例, 18
データソースの設定
ODBC データソースアドミニストレーターの使用, 39
データソースユーティリティ (dbdsn)
Vista での権限の昇格の必要性, 188
Windows 7 での権限の昇格の必要性, 188
Windows Server 2008 での権限の昇格の必要性, 188
構文, 936
システム情報ファイル, 940
終了コード, 940
データのアンロード
計算カラムの再計算, 1044
セキュリティ, 1396
文字セットの指定, 326
データのエクスポート
出力フォーマット, 898
データのロード
セキュリティ, 1396
データベース
dberase を使用した消去, 951
dbinit を使用した作成, 958
dbunload を使用したアンロード, 1044
dbupgrad を使用したアップグレード, 1061
dbvalid を使用した検証, 1064
DB 領域の削除, 126
DB 領域の作成, 123
DB 領域の変更, 125
Interactive SQL からの接続, 14
SQL Anywhere コンソールユーティリティからの接続, 14
SQL からの作成, 118
SQL からの初期化, 118
Sybase Central からの検証, 1118
Sybase Central からの作成, 118
Sybase Central からの初期化, 118
Sybase Central からの接続, 14
Sybase Central からのバックアップ, 1083
Windows Mobile 上での監査, 1409
Windows Mobile 上でのセキュリティ, 1408
Windows Mobile での Interactive SQL を使用した管理, 432
Windows Mobile での Sybase Central を使用した管理, 427

Windows Mobile での再構築, 423
Windows Mobile デバイスからの消去, 426
Windows Mobile デバイスへのコピー, 422
Windows Mobile 用に作成, 418
Windows Mobile 用に設定, 416
アルファベット順リスト, 781
暗号化, 1396
暗号化されたファイルの圧縮, 1396
アンロードのパーミッション, 236
オプションの設定, 568
既存のサービスの追加, 176
起動, 147
起動するために必要なパーミッション, 233
起動パーミッション, 233
コマンドラインからの作成, 119
コマンドラインからの消去, 141
再構築後のファイルサイズの縮小, 1056
最小サイズ, 959
サイズの変化, 1209
最大サイズ, 811
削除, 140
作成, 113
自動停止, 231
消去, 140
照合の変更, 504
情報, 957
初期化, 113
スキーマの変換, 833
正常性と統計情報, 837
設計, 113
設計上の考慮事項, 113
接続, 5
接続しないで起動, 148
接続シナリオ, 30
接続のトラブルシューティング, 76
切断, 543
タイムスタンプ, 127
大容量データベース, 121
チェックサムの検証, 1064
チュートリアル：サンプルデータベースの使用, 1
停止, 149
停止するために必要なパーミッション, 233
停止パーミッション, 235
トラブルシューティング, 1209
トランザクションログ, 113
内部アンロードのトラブルシューティング, 1060
名前の最大長, 811
名前の制限, 312
認証, 612
認証アプリケーションで使用, 181
認証データベースのアップグレード, 186
バックアップ, 1071
パーミッション, 512
比較, 832
ファイルの互換性, 113
復号化, 1401
複数のファイル, 121
フルバックアップの実行, 1074
プロパティのアルファベット順リスト, 781
ページの使用状況, 957
文字セット, 478
ユーティリティ, 135, 909
読み込み専用, 160
リカバリ, 1071
領域の割り付け, 125
ローカルデータベースへの接続, 16, 17
ロードのパーミッション, 236
データベースアクセス
制御, 1382
ユーザーが実行できるタスクの制御, 1386
データベースアップグレードウィザード
Windows Mobile でサポート対象外, 440
データベースアンロードウィザード
Windows Mobile でサポート対象外, 440
データベース移行ウィザード
Windows Mobile でサポート対象外, 440
データベースオブジェクト
DB 領域の決定, 121
SQL Anywhere 12 プラグインでコピー, 831
データベースオブジェクトのパーミッション
説明, 520
データベースオプション
ASE 互換性オプション, 585
Interactive SQL オプション, 884
Interactive SQL の設定, 884
isolation_level データベースオプション, 630
JDBC 互換性オプションのアルファベット順リスト, 585
Open Client 互換性オプションのアルファベット順リスト, 585
read_past_deleted, 669
SQL Anywhere Mobile Link クライアント用の設定, 587

SQL Anywhere SNMP Extension Agent による取得, 1328
SQL Anywhere SNMP Extension Agent による設定, 1329
SQL Anywhere SNMP Extension Agent の OID, 1362
SQL Remote オプションのアルファベット順リスト, 588
SQL Remote レプリケーションオプション, 588
SQL 文の設定, 571
Sybase Open Client, 1443
Transact-SQL 互換性オプション, 585
Transact-SQL 互換性オプションのアルファベット順リスト, 585
truncate_timestamp_values, 702
値の検索, 571
大文字と小文字の区別, 569
概要, 568
起動時の設定, 1444
初期設定, 574
スコープと継続期間, 569
設定, 568
設定の削除, 574
設定のモニタリング, 573
説明, 568
データベースオプションのアルファベット順リスト, 575
データベースサーバー (dbeng12/dbsrv12), 189
同期オプションのアルファベット順リスト, 587
分類, 575
データベースオプションの設定
説明, 568
データベース
機能の無効化, 1387
データベース管理ユーティリティ
説明, 909
データベース機能の無効化
secure_feature_key の指定, 275
secure_feature_key の使用, 677
-sf サーバーオプション, 271
説明, 1387
データベース検証ウィザード
使用, 1118
テーブルの検証, 1122
データベースサイズ
制限, 811
予期しない減少, 1209
予期しない増大, 1209
予期しない変化の解決, 1209
データベース作成ウィザード
Windows Mobile, 419
使用, 118
照合順のリスト, 959
データベースサーバー
dbping を使用して検出, 93
FIPS 認定の強力な暗号化アルゴリズムの使用, 230
LDAP の使用, 103
SATMP 環境変数, 451
UNC 接続パラメーターを使用して停止, 368
Vista での実行, 187
Windows 7 での実行, 187
Windows Mobile, 426
Windows Mobile 上で停止, 431
Windows Mobile での起動, 411
Windows Vista での自動起動, 340
オプション, 189
起動, 144
起動時のトラブルシューティング, 1211
起動の回避, 324
クワイエットモード, 160
検出, 80
コマンドライン, 189
サイレント, 160
サービス, 170
実行, 1
自動的に起動, 80
使用コア数, 245
セキュリティ, 1395
代替サーバー名の指定, 315
チュートリアル：サンプルデータベースの使用, 1
停止, 146, 280
停止するために必要なパーミッション, 235
デフォルト, 257
デフォルトにならないようにする, 292
テンポラリ接続, 76
テンポラリファイルのロケーション, 112, 451
データベース機能の無効化, 1387
データベースミラーリングによる高可用性, 1137
データベースミラーリングの優先サーバー, 1164
デーモンとして実行, 180
動作のロギング, 153

トランスポートレイヤーセキュリティを使用する起動, 1420
名前, 257
名前オプション, 151
名前のキャッシュ, 91
名前の最大長, 811
名前の制限, 257
名前のトランケーションの長さ, 258
バックグラウンドでの実行, 170
パフォーマンス警告, 1212
パーソナルサーバーとネットワークサーバーの違い, 142
プロパティ, 753
プロパティのアルファベット順リスト, 753
マルチプログラミングレベル, 165
ミラーサーバーの代替名の指定, 314
モニタリング, 1217
データベースサーバー外のアクションの監査
説明, 1395
データベースサーバーが見つからない
データベースサーバーの検出, 80
「データベースサーバーが見つかりません。」エラー
原因の診断, 95
データベースサーバーの実行
概要, 141
データベースサーバーの停止
ミラーリングシステム, 1168
データベースサーバーのモニタリング
説明, 1217
データベースサーバープロパティ
大文字と小文字の区別, 753
データベースサーバーメッセージウィンドウ
Linux での使用, 286
最大化の維持, 265
説明, 153
パフォーマンスメッセージの非表示, 266
非表示, 267
表示, 264
マルチプログラミングレベルの統計, 241
データベースサーバーメッセージログ
サイズの制限, 261
説明, 153
データベースサーバーを対象とした設定, 153
トランケート, 262
名前の取得, 258
名前の変更と再開, 260
パフォーマンス警告, 1212
ファイルの指定, 258
マルチプログラミングレベルの統計, 241
データベース消去ウィザード
Windows Mobile でサポート対象外, 440
使用, 140
データベース照合
説明, 491
データベース情報
dbinfo を使用して取得, 957
データベーススキーマ
比較, 832
比較して同じにする, 833
データベース接続
ping ユーティリティ (dbping), 994
クラウド, 24
コピーノード, 1201
読み込み専用のスケールアウト, 1201
データベース統計
SQL Anywhere MIB, 1350
SQL Anywhere SNMP Extension Agent による取得, 1328
SQL Anywhere SNMP Extension Agent の OID, 1350
データベースドキュメントウィザード
説明, 838
データベース内の Java
-cp サーバーオプション, 213
データベースの暗号化
Windows Mobile, 418
データベースのアンロード
アンロードの失敗のトラブルシューティング, 1060
アンロードユーティリティ (dbunload), 1044
計算カラムの再計算, 1044
データベースの起動
接続, 28
接続なし, 148
説明, 147
例, 145
データベースの検証
Sybase Central, 1118
検証ユーティリティの使用, 1064
説明, 1117
データベースのコピー
Windows Mobile デバイスへのコピー, 422
セキュリティについての考慮事項, 74
データベースの再構築
Windows Mobile, 423

照合の変更, 504
データベースの削除
セキュリティ, 1395
データベースの作成
Windows Mobile, 418
オプション, 959
セキュリティ, 1395
説明, 113
チュートリアル, 108
データベース作成ウィザード, 118
データベースの消去
Sybase Central, 140
Windows Mobile デバイスからの消去, 426
説明, 951
データベースの初期化
Sybase Central, 118
データベースの正常性と統計情報のモニタリング
説明, 837
データベースの設計
説明, 113
データベースの接続
説明, 5
データベースの設定
Windows Mobile, 416
データベースの停止
説明, 149
データベースのドキュメント化
説明, 838
データベースの認証
説明, 183
データベースのパーミッションと権限
説明, 512
[データベースの比較] ウィンドウ
使用, 832
データベースのモニタリング
説明, 1217
データベースバックアップウィザード
Windows Mobile でサポート対象外, 440
使用, 1083
データベースファイル
dberase を使用した消去, 951
dbinit を使用した暗号化, 959
NDS, 200
UNC ファイル名, 200
暗号化, 1396
最小サイズ, 959
最大サイズ, 811
制限, 123
セキュリティ, 1380
説明, 112
タイムスタンプ, 127
内部アンロードのトラブルシューティング, 1060
パス, 200
バックアップ, 1113
メディア障害, 1089
ロケーション, 200
データベースファイルの暗号化
初期化ユーティリティ (dbinit), 959
データベースプロパティ
SQL Anywhere SNMP Extension Agent による取得, 1328
SQL Anywhere SNMP Extension Agent による設定, 1329
SQL Anywhere SNMP Extension Agent の OID, 1355
大文字と小文字の区別, 781
レポート, 994
データベースへの BLOB の格納
説明, 116
データベースへのアクセス
セキュリティ機能, 1379
データベースへの接続
Windows Mobile, 411
データベース変更のロギングの開始
説明, 839
データベース変更のロギングの停止
説明, 839
データベースページ
キャッシュウォーミングのための収集, 207
サイズの表示, 957
データベースキャッシュの準備, 214
データベースミラーリング
LOAD TABLE 文の制限, 1140
NodeType (NODE) 接続パラメーター, 357
-sm オプション, 314
-sn オプション, 315
SQL Anywhere SNMP Extension Agent トラップ, 1331
synchronize_mirror_on_commit オプションの設定, 694
TCP/IP 接続のみサポートされる, 1140
Web サービスでの使用, 1205
Windows Mobile でサポート対象外, 435
-xa オプション, 290

-xf オプション, 292
-xp オプション, 318
監視サーバーロール, 1142
クォーラム, 1139
クライアント接続, 1163
コピーノード, 1188
サーバーの削除, 1169
サーバーの変更, 1169
システムイベント, 1173
シナリオ, 1175, 1178
ステータス情報ファイル, 1145
制限, 1140
設定, 1160
説明, 1137
チュートリアル, 1146
チュートリアル - 監視サーバーの共有, 1154
データベースサーバーの停止, 1168
同期実行モード, 1142
同期ステータス, 1144
同期モード, 1143
トランザクションログはトランケートできない, 1140
ネットワークデータベースサーバーが必要, 1140
バックアップ, 1174
パフォーマンス, 1174
パーソナルサーバーとネットワークサーバーの違い, 142
パートナーサーバー, 1138
非同期フルページモード, 1143
非同期モード, 1143
フェールオーバーの強制, 1168
プライマリサーバー障害からのリカバリ, 1170
プライマリサーバーの決定, 1164
プライマリサーバーのシャットダウン, 1166
ミラーサーバーへの読み込み専用アクセス, 1165
優先サーバー, 1164
読み込み専用のスケールアウトでの使用, 1203
利点, 1141
ログインポリシーの管理, 560
ログファイル, 1172
ロールの切り替え, 1138
データベースミラーリングで使用するモードの選択
説明, 1142
データベース名
DatabaseName 接続パラメーター, 336
大文字と小文字の区別, 152
オプション, 151
最大長, 811
設定, 312
データベースユーティリティ
Interactive SQL (dbisql), 975
ping (dbping), 994
SQL Anywhere コンソールユーティリティ (dbconsole), 1021
アップグレード (dbupgrad), 1061
アンロード (dbunload), 1044
言語選択 (dblang), 983
検証 (dbvalid), 1064
サーバー停止 (dbstop), 1024
サーバーバックグラウンド起動 (dbspawn), 1022
サーバーライセンス取得 (dblic), 1003
サーバー列挙 (dblocate), 999
サービス (dbsvc), 1013
情報 (dbinfo), 957
初期化 (dbinit), 958
スクリプト実行 (dbrunsql), 997
データソース (dbdsn), 936
データベース接続, 29
トランザクションログ (dblog), 1041
バックアップ (dbbackup), 913
パフォーマンス統計値 (dbstats), 991
バージョン診断 (dbversion) の構文, 1068
ヒストグラム (dbhist), 955
ファイル非表示 (dbfhide), 953
ログ変換 (dbtran), 985
データベースリストアウィザード
Windows Mobile でサポート対象外, 440
使用, 1091
データリカバリ
説明, 1071
テープドライブ
データベースのバックアップ, 1081
テーブル
Interactive SQL 内の検索, 857
リソース権限, 541
Sybase Central からの検証, 1122
Sybase Central でのコピー, 831
暗号化, 1405
グループ所有者, 549
修飾された名前, 562
所有者, 525
制限, 811

制約, 117
テーブル暗号化を有効にする, 1406
テーブルに必要なディスク領域の判断, 958
復号化, 1405
命名, 114
テーブル暗号化
初期化ユーティリティ (dbinit), 959
説明, 1405
テーブルサイズ
制限, 811
ローの数, 811
テーブル値
Interactive SQL での編集, 860
テーブルの検証
Sybase Central, 1122
テーブルパーミッション
設定, 526
テーブル編集の無効化
説明, 860
テーブル名
グループが所有する場合の修飾, 550
国際的な側面, 480
デーモン
-ud データベースサーバーオプション, 282
デーモンとしてのデータベースサーバーの実行, 180

## と

同期
default_timestamp_increment の設定, 619
SQL Remote delete_old_logs オプション, 621
truncate_timestamp_values の設定, 702
暗号化されたデータベースの再構築, 1061
高可用性, 1170
データベースオプション, 587
データベースミラーリング, 1142
トランスポートレイヤーセキュリティ, 1410
バックアップ, 1107
同期実行モード
データベースミラーリング, 1142
同期ステータス
データベースミラーリング, 1144
同期モード
データベースミラーリング, 1143
統計
SQL Anywhere MIB 内のデータベース統計, 1350
SQL Anywhere SNMP Extension Agent のサーバー統計 OID, 1337
[概要] タブ, 837
動的キャッシュサイズ決定, 215
統計ガバナー
update_statistics オプションを使用した無効化, 705
update_statistics オプションを使用した有効化, 705
統合化ログイン
integrated_server_name オプション, 630
login_mode オプション, 635
Windows ユーザーグループ, 57
オペレーティングシステム, 51
作成, 53
セキュリティ機能, 73
セキュリティについての考慮事項, 60
接続の禁止, 58
説明, 51
デフォルトユーザー, 60
ネットワークの局面, 59
パーミッションの取り消し, 55
有効化, 52
統合化ログインパーミッションの取り消し
説明, 55
統合データベース
高可用性, 1170
統合ログイン
クライアントアプリケーション, 56
同時接続数
最大値の設定, 237
動的キャッシュサイズ決定
最小値の設定, 211
最大値の設定, 208
データベースサーバーに対する無効化, 206
動的トラップ
説明, 1331
ドキュメント
データベースのドキュメント化, 838
特殊なグループ
説明, 551
特殊なユーザー
説明, 551
独立性レベル
設定, 630
デフォルト, 630
ミラーデータベースへの問い合わせ, 1166
ドライバー

SQL Anywhere ODBC ドライバー, 36
トラップ
SQL Anywhere SNMP Extension Agent による使用, 1330
説明, 1320
動的トラップ, 1331
トラブルシューティング
HTTP クライアント, 300
Kerberos 接続, 69
ODBC, 994
ping ユーティリティ (dbping), 994
SQL Anywhere, 1209
暗号化されたデータベースのパフォーマンス, 1403
クライアントアプリケーションの識別, 321
結果, 479
サーバーアドレス, 1442
サーバーの起動, 1211
サーバー列挙ユーティリティ (dblocate), 999
接続, 76
接続 (ネットワーク), 95
説明, 1209
タイムアウト, 1215
データベース, 1209
データベースサーバー, 299
データベースサーバーのパフォーマンス警告, 1212
データベースサーバー要求ロギング, 301
データベース接続, 76
ネットワーク通信, 1215
バックアップ, 131
プロトコル, 1215
モニター, 1317
トランケート
文字列, 691
トランザクション
イベントハンドラーの動作, 1133
カーソルを閉じる, 605
コミットされていない, 1090
データベースミラーリング, 1142
トランザクションの安全性
データベースミラーリング, 1142
データベースミラーリングでの保証, 1143
トランザクションモード
連鎖／非連鎖, 603
トランザクションログ
-ad データベースオプション, 306
-ar データベースオプション, 307
-as データベースオプション, 308
-a データベースオプション, 305
dberase を使用した消去, 951
-ds サーバーオプション, 308
-ds サーバーオプションによるリカバリ用ロケーションの指定, 308
-f リカバリオプション, 228
-m サーバーオプション, 255
-m データベースオプション, 311
SQL Remote delete_old_logs オプション, 621
SQL Remote 管理, 621
SQL Remote の自動名前変更, 1108
Windows Mobile, 417
オプション, 160
既存のデータベースでのトランザクションログミラーの開始, 130
コミットされない変更, 1090
サイズ, 130
サイズ制限, 1129
初期化ユーティリティ (dbinit), 959
推奨されるロケーション, 127
スケールアウトサーバーによる処理量, 1201
説明, 127
タイムスタンプ, 127
チェックポイント後の削除, 255
チェックポイント後のトランケート, 311
データベースミラーリング, 1172
データベースミラーリングの制限, 1140
トランザクションが複数のログファイルにまたがる場合のリカバリ, 1094
トランザクションログの検証, 1112
トランザクションログの名前を変更してバックアップ, 1110
トランザクションログミラーの作成, 120
トランザクションログミラーファイル名の設定, 1041
トランザクションログユーティリティ (dblog), 1041
配置, 1106
場所の変更, 129
バックアップコピーの名前の変更, 1109
バックアップ中の削除, 1110
バックアップユーティリティ (dbbackup), 913
バックアップ用に管理, 1115
バックアップ用に名前を変更, 1108
複数のトランザクションからのリカバリ, 1094
プライマリキー, 130
古いものの削除, 1041

変換ユーティリティ, 985
未使用での実行 (-m オプション), 959
未使用での実行に関する警告 (-m オプション), 959
未処理のトランザクションの検索, 131
メディア障害, 1100
メディア障害からのリカバリ, 1100
元のトランザクションログを使用したバックアップ, 1081
ライブバックアップ, 1075
リカバリ時におけるデータベースからのロケーションの取得, 307
リカバリ中の適用, 305
リカバリでのロケーションの指定, 306
領域の割り付け, 125
ログなしの起動, 228
ログ変換ユーティリティ (dbtran), 985
ロケーション, 200
トランザクションログの検証
説明, 1112
トランザクションログの削除
説明, 1110
トランザクションログファイル
トランザクションログユーティリティ (dblog), 1041
トランザクションログミラー
開始, 130
作成, 120
初期化ユーティリティ (dbinit), 959
推奨されるロケーション, 129
説明, 128
目的, 128
ライブバックアップとの違い, 1076
トランザクションログユーティリティ (dblog)
監査, 1395
構文, 1041
終了コード, 1044
トランスポートデータインターフェイス
サービス依存性の設定, 1016
トランスポートレイヤーセキュリティ
概要, 1410
効率, 1410
サポートされる SSL バージョン, 1425
サポートされるプラットフォーム, 1411
設定, 1412
説明, 1410
トランスポートレイヤーセキュリティの設定
説明, 1412
トランスポートレイヤーセキュリティを使用する Mobile Link クライアントの設定
説明, 1429
トランスポートレイヤーセキュリティを使用する SQL Anywhere クライアントの設定
説明, 1431
トランスポートレイヤーセキュリティを使用する Ultra Light クライアントの設定
説明, 1433
トリガー
起動不可, 235
作成パーミッション, 541
パーミッション, 528
トリガー条件
定義, 1126
トリガー
データベースドキュメントの生成, 838
レプリケーション, 624
取り消し
SQL Remote REMOTE パーミッション, 535
グループメンバーシップ, 548
権限, 532
パーミッション, 532
取り消しログ
説明, 1116
取り除く
グループからユーザーを取り除く, 548
トルコ語データベース
大文字と小文字の区別, 509
大文字と小文字を区別しないデータベース, 511
システムオブジェクトの参照, 509
ユーザー定義オブジェクトの参照, 509

## な

内部
イベント, 1132
イベント処理, 1131
イベントハンドラー, 1133
スケジュール, 1132
バックアップ, 1112
内部アンロード
使用, 1058
トラブルシューティング, 1060
夏時間
スケジュールされたイベント, 1133
ナビゲーション、プランビューアー

説明, 868
名前
接続, 332
データベース, 312
名前の変更
トランザクションログ, 1110
名前を付けた照合
データベースの作成, 503

## に

入力
Interactive SQL コマンド, 848
Interactive SQL での複数の文, 850
任意アクセス制御
ネストされたビューとテーブルの規則, 566
認証
接続, 608
データベース, 612
認証アプリケーション
connection_authentication オプション, 608
database_authentication オプション, 612
開発, 182
設定, 184
説明, 181
データベース認証, 183
認証シグネチャー, 182
プログラミングインターフェイスの例, 184
認証局
トランスポートレイヤーセキュリティ, 1419
認証シグネチャー
説明, 182
認証シグネチャーの取得
説明, 182
認証接続
connection_authentication オプション, 608
認証データベース
database_authentication オプション, 612
アップグレード, 186

## ね

ネガティブパーミッション
サポート対象外, 524
ネットワークサーバー
サポートされている通信プロトコル, 97
説明, 141
ソフトウェア要件, 142
トランスポートレイヤーセキュリティ, 1420
パーソナルサーバーとの違い, 142
マルチプログラミングレベル設定の概要, 167
マルチプログラミングレベルの設定, 237
ネットワークサーバーモニター
構文, 1021
ネットワークサーバーモニター
使用, 905
ネットワーク接続
オプション, 97
ネットワーク接続ストレージ
データベースファイルの格納, 127
ネットワーク通信
sasrv.ini ファイル, 92
起動時の問題のデバッグ, 1211
コマンドラインオプション, 369
トラブルシューティング, 1215
ネットワークドライブ
データベースファイル, 200
ネットワークプロトコル
dbeng12 -x オプション, 289
dbsrv12 -x オプション, 289
アルファベット順リスト, 369
クライアントオプション, 369
サポート, 96
説明, 96
トラブルシューティング, 1215
ネットワークプロトコルオプション
アルファベット順リスト, 369

## は

バイナリデータ型
最大サイズ, 811
ハイバネーションモード
Windows Mobile, 404
配備
Vista に関する考慮事項, 188
Windows 7 に関する考慮事項, 188
Windows Server 2008 に関する考慮事項, 188
パケットサイズ
-p dbeng12/dbsrv12 サーバーオプション, 262
-pt dbsrv12 サーバーオプション, 264
パスワード
Interactive SQL の文の履歴, 852
NEWPWD 接続パラメーター, 355
post_login_procedure オプション, 659
PWD 接続パラメーター, 360
SQL Remote での保存, 676

verify_password_function オプション, 708
暗号化, 341
インストールされるオブジェクト, 1311
検証, 1384
最小長, 648
最大長, 811
セキュリティ機能, 1383
セキュリティに関するヒント, 1380
説明, 515
デフォルト, 538
長さ, 1380
認証, 1384
変更, 515
モニターのユーザー, 1296
有効期限, 355
ユーティリティデータベース, 137
ユーティリティデータベースに設定, 276
ログインポリシー, 553
パスワード規則
login_procedure, 636
verify_password_function オプション, 708
パスワード検証
説明, 1384
パスワードの有効期限
NewPassword 接続パラメーター, 355
破損したデータベース
説明, 1117
バックアップ
BACKUP 文の使用, 1081
dbmlsync, 1107
dbremote, 1107
Mobile Link SQL Anywhere リモートデータベース, 1107
Mobile Link 統合データベース, 1115
SQL Remote, 1107
Windows Mobile での実行, 425
Windows Mobile データベース, 425
新しいトランザクションログの名前の変更と起動, 913
アーカイブ, 1077
イメージからリストア, 1091
オプション, 913
オフライン, 1071
オンライン, 1071
概要, 1071
クイックスタート, 1071
検証, 1117
コンポーネント, 1113
実行中のデータベース, 1071
実行のパーミッション, 537
自動化, 1103
終了していない, 131
種類の比較, 1072
スケジュール, 1103
制限, 1079
説明, 1071
データベースドキュメントの生成, 838
データベースのみ, 913
データベースミラーリング, 1174
テープドライブ, 1081
トランザクションログの削除, 1110
トランザクションログの名前の変更, 1108
内部, 1112
バックアップトランザクションログの名前の変更, 1109
バックアップの検証, 1087
バックアップユーティリティ (dbbackup), 913
フォーマットの選択, 1076
プラン, 1103
フル, 1074
並列, 1116
元のトランザクションログを使用, 1081
モニター, 1309
ライブ, 1075
リモートデータベース, 1107
レプリケーションに関連しないデータベース, 1115
バックアップイメージ作成ウィザード
使用, 1083
バックアップ権限
オンラインバックアップに必要, 1071
継承不可能, 537
説明, 537
付与, 542
バックアップディレクトリ
バックアップユーティリティ (dbbackup), 913
バックアップとリカバリ
Windows Mobile での方法, 425
概要, 1071
バックアップの自動化
説明, 1103
バックアップフォーマット
種類, 1076
バックアッププラン
説明, 1101
データベースのドキュメント化, 838

バックアップユーティリティ (dbbackup)
エラーの受信, 913
クライアント側バックアップ, 1085
構文, 913
終了コード, 920
バックグラウンド
データベースサーバーの実行, 170
バックスラッシュ
データベースサーバー名に使用できない, 257
バッチファイル
dbspawn によるデータベースサーバーの起動, 200
サーバーを起動, 1022
パフォーマンス
Embedded SQL 接続のテスト, 93
OLAP クエリ, 658
Sybase PowerBuilder DataWindow, 655
TCP/IP, 95
圧縮, 106
暗号化されたデータベース, 1403
キャッシュサイズ, 156
結果セット, 655
サーバーオプション, 156
ディスクの断片化, 125
データベースサーバーメッセージログ内の警告, 1212
データベースサーバーメッセージログ内のメッセージ, 1212
データベースミラーリング, 1174
テーブル暗号化が及ぼす影響, 1406
トランザクションログサイズ, 130
トランザクションログの効果, 127
トランザクションログミラー, 128
プライマリキー, 130
プリフェッチ, 662
優先度の設定, 664
パフォーマンス警告
説明, 1212
パフォーマンス統計値
UNIX でのモニタリング, 991
接続の無効化, 249
パフォーマンス統計値ユーティリティ (dbstats)
構文, 991
終了コード, 993
パフォーマンスの向上
適切なハードウェアの入手, 814
パブリックキー暗号方式
説明, 1410
パブリックキーインフラストラクチャーオブジェクト
表示, 934
バルクオペレーション
-b サーバーオプション, 203
バルクロード
オプション, 160
バージョン
データベースサーバー, 287
判別, 1068
バージョン診断 (dbversion)
構文, 1068
バージョンの不一致
ファイルロケーション, 460
パーシートライセンス
説明, 1003
パーソナルサーバー
TCP/IP の起動, 289
Windows Mobile でサポート対象外, 435
サポートされている通信プロトコル, 97
説明, 141
ネットワークサーバーとの違い, 142
マルチプログラミングレベル設定の概要, 167
マルチプログラミングレベルの設定, 237
パーソナルサーバーサンプル
実行, 1
ハードウェア
要件, 814
ハードウェアミラーリング
トランザクションログ, 129
パートナーサーバー
説明, 1137
データベースミラーリングの定義, 1160
パーミッション
DBA, 538
DB 領域, 528
READCLIENTFILE 権限, 540
READFILE 権限, 540
REMOTE DBA 権限, 540
リソース権限, 541
SQL Remote REMOTE の取り消し, 535
SQL Remote REMOTE の付与, 535
VALIDATE 権限, 541
WITH GRANT OPTION, 531
WRITECLIENTFILE 権限, 541
オプション, 159
管理, 512
グループ, 544

グループメンバーシップ, 547
継承, 525, 531
高度なセキュリティを実現するためにビューを使用, 564
個別, 512
サポート対象外のネガティブパーミッション, 524
システムビュー, 518
シーケンスジェネレーター, 528
スキーム, 1382
セキュリティ機能, 1382
説明, 520
テンポラリファイル, 452
データのアンロード, 236
データのロード, 236
データベースサーバーの停止, 235
データベースの起動, 233
データベースの停止, 233
テーブル, 526
統合化ログインパーミッション, 51
取り消し, 532
パスワード, 515
バックアップ権限, 537
ビューに対する付与, 527
ファイル管理文, 561
付与権, 531
プロシージャー, 527
プロシージャーに対する設定, 527
プロファイル権限, 539
無効化されたオブジェクト, 526
矛盾, 545

## ひ

非書き込みモード
データベースサーバー, 247
比較
TIMESTAMP, 619
データベーススキーマ, 832
バックアップの種類, 1072
ヒストグラム
dbhist を使用して表示, 955
ヒストグラムユーティリティ (dbhist)
構文, 955
終了コード, 956
日付
date_order オプション, 615
nearest_century オプション, 649
非同期 I/O
Linux での使用の無効化, 281
非同期フルページモード
データベースミラーリング, 1143
非同期モード
データベースミラーリング, 1143
ビュー
リソース権限, 541
Sybase Central でのコピー, 831
所有者, 525
セキュリティ, 564
セキュリティ機能, 1379
データベースドキュメントの生成, 838
パーミッションの付与, 527
表示
TLS 証明書, 934
ひらがな
照合の適合化, 496
ビルド番号
SQL Anywhere, 287
非連鎖モード
chained オプション, 603

## ふ

ファイアウォール
BroadcastListener (BLISTENER) プロトコルオプション, 373
ClientPort (CPORT) プロトコルオプション, 377
Host (IP) プロトコルオプション, 381
LDAP との接続, 103
LDAP プロトコルオプション, 387
ServerPort (PORT) プロトコルオプション, 396
サーバー列挙ユーティリティ (dblocate), 1002
接続, 101
ファイル
Interactive SQL からソース制御を使用, 873
Interactive SQL からの更新, 877
Interactive SQL からのチェックアウト, 876
Interactive SQL からのチェックイン, 877
Interactive SQL ソース制御を設定, 874
暗号化, 1398
ロケーション, 460
ファイルサイズ
再構築後の縮小, 1056
ファイルデータソース
作成, 45
ファイルのソース指定

UNIX/Mac, 444
ファイルのロケーション
Windows Mobile, 459
ファイル非表示ユーティリティ (dbfhide)
構文, 953
ファンクションキー
Interactive SQL, 878
フェールオーバー
Veritas Cluster Server と SQL Anywhere, 1182
クラスタ, 1182
データベースミラーリング, 1137
フェールオーバー
データベースミラーリングのシナリオ, 1175, 1178
フォルダーリスト
Sybase Central の表示, 820
フォローバイト
接続文字列, 478
説明, 475
フォワードスラッシュ
データベースサーバー名に使用できない, 257
フォント
コードエディターの設定, 825
フォーマット
入力ファイル, 890
復号化
テクニカルサポートのためのデータベース, 1401
データベース, 1401
テーブル, 1405
複数オプションキーワード
オフにできない, 650
複数の結果セット
Interactive SQL オプション, 896
isql_show_multiple_result_sets, 896
複数のデータベース
DSEdit エントリ, 1440
物理的な制限事項
SQL Anywhere, 811
付与
SQL Remote REMOTE パーミッション, 535
プライベートキー
表示, 934
プライマリサーバー
決定, 1164
コピーノードとの違い, 1204
障害からのリカバリ, 1170
スケールアウトシステムからのサーバーの削除, 1202
停止, 1168
データベースミラーリングの概要, 1137
フェールオーバーの強制, 1168
ミラーサーバーの削除, 1169
プライマリサーバー障害からのリカバリ
説明, 1170
プラグイン
SQL Anywhere, 830
レジストリ設定, 466
ブラックアウト
説明, 1260
プラン
max_plans_cached オプション, 643
オプティマイザーによる使用を制御, 657
最近のプランの取得, 301
バックアップ, 1103
バックアップとリカバリ, 1101
ブランク
ANSI の性質, 592
ブランク埋め込み
初期化ユーティリティ (dbinit), 959
説明, 959
プランビューアー
ファイル拡張子, 867
プランビューアー
説明, 867
ナビゲーション, 868
プライマリキー
トランザクションログ, 130
フルバックアップ
説明, 1074
フレームタイプ
説明, 95
プロキシポート
Windows Mobile デバイス用に作成, 413
プログラミングインターフェイス
接続, 49
プロシージャー
作成パーミッション, 541
セキュリティ, 564
パーミッション, 527
プロシージャー
Interactive SQL 内の検索, 857
max_plans_cached オプション, 643
データベースドキュメントの生成, 838
プロセッサー

使用する数, 244
複数, 158
プロセッサーライセンス
説明, 1003
プロセッサー
同時実行性, 245
プロトコル
TCP/IP を使用したデータベースサーバー, 370
オプション, 97
サポート, 96
説明, 96
選択, 97
データベースサーバー用の指定, 290
トラブルシューティング, 1215
プロトコルオプション
HTTPS を使用したデータベースサーバー, 370
HTTP を使用したデータベースサーバー, 370
TLS を使用したデータベースサーバー, 370
アルファベット順リスト, 369
大文字と小文字を区別しない, 370
ブール値, 370
リスト, 370
プロパティ
SQL Anywhere MIB 内のサーバープロパティ OID, 1341
SQL Anywhere MIB 内のデータベースプロパティ, 1355
サーバープロパティのアルファベット順リスト, 753
接続, 713
接続プロパティのアクセス, 713
接続プロパティのアルファベット順リスト, 713
データベース, 781
データベースサーバー, 753
データベースサーバープロパティのアクセス, 753
データベースプロパティのアクセス, 781
データベースプロパティのアルファベット順リスト, 781
プロファイリング
実行のパーミッション, 539
プロファイル権限
継承可能, 539
説明, 539
付与, 542
ブロードキャストプロトコルオプション
IPv6 アドレスの使用, 101
文
Interactive SQL でのキャンセル, 857
Interactive SQL での再呼び出し, 852
Interactive SQL での実行, 848
Interactive SQL での中断, 857
Interactive SQL での停止, 857
Interactive SQL での編集, 852
Interactive SQL でのロギング, 856
Windows Mobile でサポート対象外の文, 436
クライアントでのキャッシュ, 641
制限, 811
データベースオプション, 571
ユーティリティデータベース, 135
ロギング, 154
分散トランザクション
-tmf オプションを使用したリカバリ, 279
エンリストのタイムアウト, 280
分散トランザクションコーディネーター
使用しないでリカバリ, 279
文の実行
認証文, 184
プール
ConnectionPool (CPOOL) 接続パラメーター, 332
SQL Anywhere での接続, 74
ブール値
接続パラメーター, 321
プロトコルオプション, 370


## へ


並列実行
プロセッサー, 244
並列処理
max_query_tasks オプション, 644
並列バックアップ
dbbackup ユーティリティ, 913
Windows Mobile でサポート対象外, 435
説明, 1116
[別のコンピューターで稼働しているデータベースに接続]
例, 22
[別のコンピューターで稼働しているデータベースに接続]
[接続] ウィンドウ, 14
別のコンピューターのデータベースを起動して接続
[接続] ウィンドウ, 14

例, 25
変換
PKI オブジェクトのコード化, 934
データベーススキーマ, 833
編集
Interactive SQL での結果セット, 859
接続プロファイル, 12
ページ
書き込みチェックサム, 1120
データベースファイル内での使用状況の表示, 957
トランザクションログ, 127
ページサイズ
オプション, 150
許容最大数, 242
選択, 959
データベース, 958
ページの使用状況
情報ユーティリティ (dbinfo), 957

## ほ

保護された機能
secure_feature_key オプション, 677
-sf での指定, 271
-sk での secure_feature_key の指定, 275
説明, 1387
ホストプロトコルオプション
IPv6 アドレスの使用, 101
ポリシー
SELinux のサポート, 1380
SQL Anywhere ログイン, 553
ポリシーオプション
について, 553
ポート
デフォルト, 396
ポート番号
ServerPort (PORT) プロトコルオプション, 396
高速ランチャー, 903
データベースサーバー, 396

## ま

前のスペース
接続文字列での使用, 31
マテリアライズドビュー
materialized_view_optimization オプション, 639
max_client_statements_cached オプション, 640
マネージャー
説明, 1320
マルチキャストアドレス
IPv6 サポート, 101
マルチタスク
スレッドの制御, 161
マルチバイト文字セット
使用, 487
説明, 475
マルチプログラミングレベル
自動チューニング, 165
自動調整情報の表示, 241
手動調整, 169
増加, 168
チューニング, 166
低下, 168
データベースサーバー, 165
動的チューニング, 238
マルチプロセッサーサポート
サーバーオプション, 156
スレッドの制御, 161
マルチプロセッシング
スレッドの制御, 161
丸め
scale オプション, 676

## み

未加工プロパティ
説明, 810
未送信のエラーレポート
表示, 1210
モニターの警告, 1304
未送信のエラーレポートの抑制
モニター, 1304
ミラー
トランザクションログ, 128
ミラーサーバー
コピーノードとの違い, 1204
削除, 1169
停止, 1168
データベースミラーリングの概要, 1137
読み込み専用のデータベースアクセス, 1165
ミラーデータベース
問い合わせ, 1166
読み込み専用アクセス, 1165
ミラーリング
Mobile Link, 1170
-sm オプション, 314

-sn オプション, 315
SQL Anywhere SNMP Extension Agent トラップ, 1331
synchronize_mirror_on_commit オプションの設定, 694
Web サービスでの使用, 1205
-xa オプション, 290
-xf オプション, 292
-xp オプション, 318
監視サーバーロール, 1142
クライアント接続, 1163
システムイベント, 1173
シナリオ, 1175, 1178
ステータス情報ファイル, 1145
制限, 1140
設定, 1160
説明, 1137
チュートリアル, 1146
チュートリアル - 監視サーバーの共有, 1154
データベースサーバーの停止, 1168
同期実行モード, 1142
同期ステータス, 1144
同期モード, 1143
バックアップ, 1174
パフォーマンス, 1174
非同期フルページモード, 1143
非同期モード, 1143
プライマリサーバー障害からのリカバリ, 1170
プライマリサーバーの決定, 1164
プライマリサーバーのモニタリング, 1249
利点, 1141
ミラーリングシステム
コピーノードの親の特定, 1200
説明, 1137
民間認証局
トランスポートレイヤーセキュリティ, 1414

## む

無効化
Interactive SQL でのテーブル編集, 861

## め

明示的な選択性推定
user_estimates オプション, 706
命名
接続, 332
テーブルとカラム, 114
メタデータテーブル
SQL Anywhere MIB, 1334
メッセージ
言語リソースライブラリ, 474
データベースサーバーのパフォーマンス, 1212
マルチプログラミングレベルの変更, 241
メッセージリンクパラメーター
SQL Remote の external_remote_options, 623
メッセージログ
説明, 153
データベースサーバーのパフォーマンス警告, 1212
データベースサーバーを対象とした設定, 153
モニター, 1248
メディア障害
説明, 1106
トランザクションログ, 1100
保護, 1106
メトリック
モニター, 1262
モニターのメトリックリスト, 1265
メモリ
AWE キャッシュサイズの制限, 212
最小キャッシュサイズの設定, 211
最大キャッシュサイズの設定, 208
初期キャッシュサイズの設定, 204
静的キャッシュサイズの設定, 206
接続制限, 552
要件, 814
メモリカード
Windows Mobile, 404
メンテナンスプラン
ステータス, 1104
説明, 1104
レポート, 1105
メンテナンスプラン作成ウィザード
Windows Mobile で一部サポート, 440
使用, 1105
メンテナンスプランレポート
説明, 1105
メンバーシップ
グループメンバーシップの取り消し, 548

## も

文字
照合を使用したソート, 487
文字 V と W

スウェーデン語の UCA 照合での識別, 496
文字コード
説明, 473
文字セット
ASE ラベル, 476
CHAR 文字セットの確認, 501
IANA ラベル, 476
IANA ラベルリスト, 507
ICU ラベル, 476
Java ラベル, 476
Mac OS X の推奨, 506
MIME ラベル, 476
NCHAR 文字セットのかくにん, 501
SQL Anywhere 内, 476
Unicode, 487
UNIX のデフォルト, 487
UNIX プラットフォームに対する推奨, 508
Windows, 476
Windows のデフォルト, 487
Windows プラットフォームでの推奨, 507
アプリケーション, 487
可変幅, 475
固定幅, 475
コード化, 468
サーバー, 487
指定, 449
初期化ユーティリティ (dbinit), 959
シングルバイト, 474
接続パラメーター, 325
説明, 473
代替エンコード, 476
提供されている CHAR エンコード, 476
データのアンロード, 326
トルコ語データベース, 509
変換, 478
マルチバイト, 475
マルチバイト照合, 487
文字セットのサポート状況の確認, 476
ラベル, 507
文字セット変換
ICU, 478
SQL 文, 479
クライアント API, 480
クライアント／サーバー, 479
説明, 478
文字の置換
on_charset_conversion_failure オプション, 654
文字列
最大サイズ, 811
ホスト変数, 592
モニター
admin ユーザー, 1297
インストールされているオブジェクトの削除, 1312
インストールされるオブジェクト, 1311
エラーレポート, 1304
起動, 1235
警告, 1301
終了, 1238
制限, 1220
説明, 1217
チュートリアル, 1222
電子メールによる通知, 1307
トラブルシューティング, 1317
別のコンピューターにインストール, 1312
メトリック, 1262
モニターの状態, 1241
モニターの停止, 1238
ユーザータイプ, 1296
ユーザーとリソースの関連付け, 1299
要件, 1219
リソースのモニタリングの停止, 1258
モニターの状態
モニター, 1241
モニタリング
Mobile Link サーバー, 1217
Relay Server、説明, 1217
データベース, 1217
データベースオプション設定, 573
ヒストグラムを使用して表示, 955
モニター, 1217
ログオンしているユーザー, 1021
モニター
[ホスト] フィールド, 1249
ECC, 1316
FIPS, 1316
Mobile Link、samonitor.bat, 1236
Mobile Link サーバーファームメトリックリスト, 1281
Mobile Link、説明, 1217
Mobile Link の状態, 1241
Mobile Link、メトリック, 1262
Mobile Link、メトリックのエクスポート, 1263
Mobile Link、メトリックリスト, 1270
Mobile Link リソース, 1249
Mobile Link、リソースのインポート, 1252

Mobile Link、リソースのモニタリングの停止, 1258
Relay Server、samonitor.bat, 1236
Relay Server、説明, 1217
Relay Server の移行, 1314
Relay Server の状態, 1241
Relay Server ファームメトリックリスト, 1294
Relay Server、メトリック, 1262
Relay Server、メトリックのエクスポート, 1263
Relay Server リソース, 1249
Relay Server、リソースのインポート, 1252
Relay Server、リソースのモニタリングのていし, 1258
samonitor.bat, 1236
TLS, 1316
ウィジェット, 1244
運用環境, 1312
[概要] ダッシュボード, 1240
[管理] ウィンドウ, 1247
警告の重大度, 1242
時刻, 1247
接続の切断, 1247
ダッシュボード, 1243
デフォルトユーザー, 1297
バックアップ, 1309
ブラックアウト, 1260
メッセージログ, 1248
メトリックのエクスポート, 1263
メトリックのリフレッシュ, 1264
メトリックリスト, 1265
リソース, 1249
リソースのインポート, 1252
例外レポート, 1248
ログアウト, 1240
ログイン, 1239
モニターの状態
Mobile Link モニター, 1241
Relay Server モニター, 1241
モード
SQL Anywhere 12 プラグイン, 830
データベースミラーリングでの同期実行, 1142

## ゆ

有効化
テーブル暗号化, 1406
優先サーバー
データベースミラーリングに指定, 1164
優先度
接続パラメーター, 35
プロセス, 232
ユーザー ID
システムビュー, 518
ユーザー
Kerberos ログインの削除, 68
Kerberos ログインの作成, 67
SQL Remote REMOTE パーミッション, 535
オプションの設定, 516
管理, 512
グループから取り除く, 548
グループへの追加, 547
削除, 517
作成, 513
接続しているユーザー, 542
追加, 513
デフォルト, 513
テンポラリ領域の制限, 647
統合化ログインの削除, 55
統合化ログインの付与, 51
パーミッション, 512
パーミッションの矛盾, 545
モニターのタイプ, 1296
モニターのユーザー, 1297
ログインポリシー, 553
ログインポリシーからの削除, 558
ログインポリシーの割り当て, 558
ユーザー ID
DBA, 538
Guest, 60
PUBLIC オプション, 570
管理, 512
個別のユーザー ID の設定, 512
最大長, 811
セキュリティ機能, 1379
セキュリティに関するヒント, 1380
説明, 512
ログインポリシー, 553
ユーザーアカウント制御
Vista での SQL Anywhere の実行, 187
Windows 7 での SQL Anywhere の実行, 187
Windows Server 2008 での SQL Anywhere の実行, 187
ユーザーが実行できるタスクの制御
説明, 1386
ユーザーが提供する選択性推定
user_estimates オプション, 706

ユーザー作成ウィザード
使用, 514
ユーザー推定
上書き, 706
ユーザー設定
SQL Anywhere コンソールユーティリティ (dbconsole), 906
ユーザー定義プロパティ
説明, 810
用途, 810
ユーザーの作成
説明, 557
ユーザー名
サーバーライセンス取得ユーティリティ (dblic), 1003
ユーティリティ
Broadcast Repeater (dbns12) の構文, 921
dbisqlc 構文, 949
DSEdit, 1438
Interactive SQL (dbisql) 構文, 975
Linux サービス (dbsvc) の構文, 1007
ping (dbping) の構文, 994
SQL Anywhere コンソール (dbconsole) の構文, 1021
SQL Anywhere スクリプト実行ユーティリティ (dbrunsql) の構文, 997
UNIX 上のソース, 444
Windows サービス (dbsvc) の構文, 1013
アップグレード (dbupgrad) の構文, 1061
アンロード (dbunload) の構文, 1044
概要, 909
キーペアジェネレーター (createkey) の構文, 981
言語選択 (dblang) の構文, 983
検証 (dbvalid) の構文, 1064
サポート (dbsupport) の構文, 1027
サポートされるプラットフォーム, 909
サーバー停止 (dbstop) の構文, 1024
サーバーバックグラウンド起動 (dbspawn) の構文, 1022
サーバーライセンス取得 (dblic) の構文, 1003
サーバー列挙 (dblocate) の構文, 999
消去 (dberase) の構文, 951
情報 (dbinfo) の構文, 957
証明書作成 (createcert) の構文, 924
証明書ビューアー (viewcert) の構文, 934
初期化 (dbinit) の構文, 958
設定ファイルでの条件付き解析の使用, 910
設定ファイルの使用, 909
デバイスでの Windows Mobile データベース管理, 427
データソース (dbdsn) の構文, 936
トランザクションログ (dblog) の構文, 1041
認証アプリケーションで使用, 182
バックアップ (dbbackup) の構文, 913
パフォーマンス統計値 (dbstats) の構文, 991
バージョン診断 (dbversion) の構文, 1068
ヒストグラム (dbhist) の構文, 955
ファイル非表示 (dbfhide) の構文, 953
ログ変換 (dbtran) の構文, 985
ユーティリティコマンド
パーミッション, 247
ユーティリティデータベース
util_db.ini ファイル, 139
使用できる SQL 文, 136
進行メッセージ, 666
セキュリティ, 561
接続, 137
説明, 135
パスワードの設定, 276
パーソナルサーバーとネットワークサーバーの違い, 142
文実行パーミッションの制御, 561

## よ

要求
request_timeout オプション, 672
SQL Anywhere でのスレッド, 161
要求ログ
コピー数, 298
要求ロギング
max_client_statements_cached の設定, 642
データベースサーバーオプション, 301
ファイルへのログ情報の保存, 299
要求ログのコピー数, 298
ログファイルサイズの制限, 303
要求ログ
max_client_statements_cached の設定, 642
サイズ制限, 303
使用, 299
要件
SQL Anywhere for Windows Mobile, 404
ハードウェア, 814
予測
リカバリ時間, 669

ローカウント, 675
読み込み
TLS 証明書, 934
読み込み専用
-sm オプション, 314
データベース, 160
ミラーデータベースへのアクセス, 1165
読み込み専用データベース
-r dbeng12/dbsrv12 サーバーオプション, 267
-r dbeng12/dbsrv12 データベースオプション, 313
スケールアウト, 560
データベースミラーリング, 560
読み込み専用のスケールアウト
NodeType (NODE) 接続パラメーター, 357
SQL Anywhere モニターからのモニタリング, 1217
-xp オプション, 318
コピーノードの定義, 1198
サーバーの削除, 1202
接続プールでの使用, 75
説明, 1188
チュートリアル, 1190
データベースミラーリングでの使用, 1203
パーソナルサーバーとネットワークサーバーの違い, 142

## ら

ライセンス
-gm サーバーオプションへの影響, 237
-gtc サーバーオプションへの影響, 245
-gt サーバーオプションへの影響, 244
サーバーライセンス取得ユーティリティ (dblic) を使用した追加, 1003
実行プログラム, 1005
スレッドの効果, 164
接続制限とイベントハンドラー, 1133
パーシートライセンス, 1003
パーソナルサーバーとネットワークサーバーの違い, 142
プロセッサーライセンス, 1003
ライセンス取得
サーバーライセンス取得 (dblic) の構文, 1003
ライセンスタイプ
サーバーライセンス取得ユーティリティ (dblic), 1003
ライセンスファイル
説明, 1005
ライブバックアップ
バックアップユーティリティ (dbbackup), 913
ライブバックアップ
作成, 1086
説明, 1075
トランザクションログミラーとの違い, 1076
ライブバックアップとトランザクションログミラーの違い
説明, 1076
ライブバックアップの作成
説明, 1075
ライブラリ
dbping ユーティリティのロード, 994
DYLD_LIBRARY_PATH 環境変数 [Mac OS X], 445
Kerberos GSS-API ライブラリファイル, 63
Kerberos 認証, 62
LD_LIBRARY_PATH 環境変数 [Linux と Solaris], 445
LIBPATH 環境変数 [IBM AIX], 446
SHLIB_PATH 環境変数 [HP-UX], 453
Windows Mobile での ICU の使用に必要, 404
インターフェイスライブラリの検出, 77
ラベル
言語ラベルの値, 485
文字セット, 507

## り

リカバリ
Windows Mobile での方法, 425
オプション, 160
緊急度, 1114
高速, 1075
コミットされない変更, 1090
最大時間, 242
サーバーオプション, 228
システム障害, 1106
自動, 1071
制限, 1079
説明, 1071
データベース, 1089
トランザクションログ, 1100
トランザクションログミラー, 128
トランザクションログロケーションの指定, 308
内部, 1112

分散トランザクションオプション, 279
メディア障害, 1097
リスト
データベースオプションのモニタリング, 573
リソース
Mobile Link のアイコンの説明, 1241
MobiLink リソースのインポート, 1252
Relay Server のアイコンの説明, 1241
Relay Server リソースのインポート, 1252
アイコンの説明, 1241
モニター、修復, 1260
モニター、リソースのインポート, 1252
リソースガバナー
カーソル, 642
定義, 553
文, 646
リターンコード
Interactive SQL ユーティリティ (dbisql), 981
ping ユーティリティ (dbping), 997
Windows サービスユーティリティ (dbsvc), 1020
アップグレードユーティリティ (dbupgrad), 1064
アンロードユーティリティ (dbunload), 1058
言語ユーティリティ (dblang), 984
検証ユーティリティ (dbvalid), 1068
サーバー停止ユーティリティ (dbstop), 1026
サーバーバックグラウンド起動ユーティリティ (dbspawn), 1024
サーバーライセンス取得ユーティリティ (dblic), 1006
サーバー列挙ユーティリティ (dblocate), 1002
消去ユーティリティ (dberase), 953
情報ユーティリティ (dbinfo), 958
初期化ユーティリティ (dbinit), 975
データソースユーティリティ (dbdsn), 940
トランザクションログユーティリティ (dblog), 1044
バックアップユーティリティ (dbbackup), 920
パフォーマンス統計値ユーティリティ (dbstats), 993
ヒストグラムユーティリティ (dbhist), 956
ログ変換ユーティリティ (dbtran), 991
利点
データベースミラーリング, 1141
リフレッシュ
モニター, 1264
リモートデータアクセス
CIS_ OPTION オプション, 604
cis_rowset_size オプション, 605
Windows Mobile でサポート対象外, 435
リモートデータベース
高可用性, 1170

## る

ルータ
同時送信, 379
ルート証明書
トランスポートレイヤーセキュリティ, 1414
トランスポートレイヤーセキュリティのクライアント検証, 1419
ルートログインポリシー
説明, 553
変更, 556
ルートログインポリシーの変更
説明, 556

## れ

例外レポート
モニター, 1248
レジストリ
SQLREMOTE 環境変数の設定, 456
Sybase Central, 466
Windows Mobile, 466
Windows サービス, 464
環境変数, 443
言語設定, 466
言語選択ユーティリティ (dblang), 983
修正, 443
説明, 463
ツールロケーション設定, 466
ロケーション設定, 466
レプリケーション
SQL Remote オプション, 588
暗号化されたデータベースの再構築, 1061
トランザクションログの管理, 1107
トリガーアクション, 624
バックアッププロシージャー, 1107
レプリケーションオプション
SQL Remote blob_threshold, 600
SQL Remote compression, 607
SQL Remote delete_old_logs, 621
SQL Remote external_remote_options, 623
SQL Remote qualify_owners, 667
SQL Remote quote_all_identifiers, 668

SQL Remote replication_error, 671
SQL Remote replication_error_piece, 671
SQL Remote save_remote_passwords, 676
SQL Remote sr_date_format, 681
SQL Remote sr_time_format, 682
SQL Remote sr_timestamp_format, 683
SQL Remote sr_timestamp_with_time_zone_format, 684
SQL Remote subscribe_by_remote, 692
SQL Remote verify_all_columns, 707
SQL Remote verify_threshold, 711
SQL Remote リスト, 588
初期設定, 574
分類, 575
レベル 4 句読表記の区別
大文字小文字とアクセント記号を区別しないデータベース, 974
レポート
メンテナンスプラン, 1105
連鎖トランザクションモード
chained オプション, 603
連邦情報処理規格
説明, 1412

## ろ

ロギング
HTTP クライアント情報, 300
HTTP サーバー要求ロギング, 388
HTTP ログフォーマット, 389
Interactive SQL 内のコマンド, 856
データベースサーバーの動作, 153
データベースサーバーメッセージ, 258
データベース変更, 839
トランザクションログ, 127
ログイン
Kerberos, 61
失敗 DBA, 538
統合化, 51
ログインプロシージャー
Web サービスで実行, 637
説明, 636
ログインポリシー
DBA, 538
新しいユーザー作成時の割り当て, 557
既存のユーザーへの割り当て, 558
削除, 560
作成, 557
説明, 553
変更, 559
読み込み専用データベース, 560
ルートポリシーからの継承, 555
ルートポリシーのポリシーオプションの上書き, 553
ルートログインポリシー, 553
ログインポリシーからのユーザーの削除
説明, 558
ログインポリシー作成ウィザード
使用, 557
ログインポリシーの管理
説明, 553
読み込み専用データベース, 560
ログインポリシーの削除
説明, 560
ログインポリシーの作成
説明, 557
ログインポリシーの変更
説明, 559
ログインポリシーの割り当て
説明, 558
ログインマッピング
Kerberos, 61
統合化ログイン, 51
ログインマッピング作成ウィザード
使用, 53
ログインモード
login_mode オプション, 635
Windows 統合化ログイン, 51
ログオフ
サーバーの実行を維持, 282
ログビューアー
説明, 829
開く, 830
ログファイル
echo オプション, 889
監査, 1395
チェックポイントログ, 133
データベースサーバーメッセージログ, 153
データベースミラーリング, 1172
トランザクション, 127
トランザクションログミラー, 128
トランザクションログユーティリティ (dblog), 1041
ロールバック, 1116
ログファイル設定の変更ウィザード
Windows Mobile でサポート対象外, 440

既存のデータベースでのトランザクションログミラーの開始, 130
使用, 129
ログファイル変換ウィザード
Windows Mobile でサポート対象外, 440
使用, 1090
ログ変換ユーティリティ (dbtran)
監査, 1395
監査情報の取り出し, 1391
構文, 985
コミットされていない操作のリカバリ, 1090
終了コード, 991
使用, 1094
ロケール
決定, 501
言語, 485
設定, 502
説明, 484
文字セット, 487
文字セットの変換, 478
ロケール定義
説明, 484
ロック
ミラーデータベース, 1166
ロック競合
blocking_timeout オプション, 602
blocking オプション, 601
ロー
Interactive SQL での値の編集, 860
Interactive SQL でのコピー, 862
Interactive SQL での挿入, 861
Interactive SQL を使用した削除, 862
Interactive SQL を使用した追加, 861
ローカウント
有効, 675
ローカルマシン
環境設定, 464
ロード
READCLIENTFILE 権限, 540
ロール
データベースミラーリング, 1138
ロールの切り替え
データベースミラーリング, 1138
ロールバックログ
説明, 1116
データベースサイズの増大, 1209

## わ

割り当て
ログインポリシー、新しいユーザー作成時, 557
ワーカー
Linux の動作, 163
SQL Anywhere でのスレッド, 161
UNIX の動作, 163
Windows の動作, 163
スタックサイズ, 243