

# アプリケーションでのデータベース・サーバとデータベースの検出

**概要:**ここでは、dbping 機能をアプリケーションに統合する手順を説明します。dbping は、データベース・サーバや特定の SQL Anywhere データベースが存在するかどうかを検出します。

## バックグラウンド情報

dbping ユーティリティは、データベース・サーバやそのデータベースが存在するかどうかを検出するために使用するコマンド・ライン・ユーティリティです。次に、サーバを検出するコマンドの例を示します。

```
dbping -c "uid=dba;pwd=sql;eng=servername"
```

次に、特定のデータベースを検出するコマンドの例を示します。

```
dbping -c "uid=dba;pwd=sql;eng=servername;dbn=databasename" -d
```

アプリケーションで同様のデータベース検出機能を使用する場合は、コマンド・ライン・ユーティリティは適していません。ここでは、アプリケーションで同様の機能を実現する手順を簡単に説明します。

dbping の詳細については、SQL Anywhere Studio 7.0.3 のマニュアルの次のトピックを参照してください。

- 『ASA ユーザーズ・ガイド』
- 「第 1 部 データベースの起動と接続」
- 「第 2 章 データベースへの接続」
- 「接続のトラブルシューティング」
- 「サーバを見つけられるかどうかのテスト」
  
- 『ASA リファレンス・マニュアル』
- 「第 4 章 データベース管理ユーティリティ」
- 「Ping ユーティリティ」

## Embedded SQL アプリケーション

次の例は、servername というサーバが存在するかどうかを検出する dbping コマンドを、Embedded SQL で実現したものです。

```
EXEC SQL INCLUDE sqlca;  
bool success;  
db_init( &sqlca );  
success = db_string_ping_server( &sqlca, "uid=dba;pwd=sql;eng=servername", FALSE );  
...  
db_fini( &sqlca );
```

この例は、databasename というデータベースが存在するかどうかを検出する dbping コマンドを、Embedded SQL で実現したものです。

```

EXEC SQL INCLUDE sqlca;
bool success;
db_init( &sqlca );
success = db_string_ping_server( &sqlca,
"uid=dba;pwd=sql;eng=servername;dbn=databasename", TRUE );
...
db_fini( &sqlca );

```

db\_string\_ping\_server() 関数は、ASA の h サブフォルダ内にある sqlfunc.h で定義されています。詳細については、SQL Anywhere Studio 7.0.3 のマニュアルの次の項を参照してください。

- 『ASA プログラミング・インターフェース』
- 「第 2 章 Embedded SQL インターフェース」
- 「ライブラリ関数」
- 「接続およびサーバ管理関数」
- 「db\_string\_ping\_server 関数」

## ODBC アプリケーション

ODBC には、次のコマンド・ラインに相当するものではありません。

```
dbping -c "uid=dba;pwd=sql;eng=servername"
```

ユーザがユーティリティ・データベースにパスワードを設定している場合は、指定したサーバの utility\_db に接続しようとするのが、ODBC でのサーバ検出に相当します。これは、以下で説明するように、データベース・サーバを実行するデータベースを検出するのと同様の状況になります。ユーティリティ・データベースの詳細については、SQL Anywhere Studio 7.0.3 のマニュアルの次の項を参照してください。

- 『ASA ユーザーズ・ガイド』
- 「第 5 部 データベースの管理と高度な使用方法」
- 「第 2 5 章 データベース・ファイルの処理」
- 「ユーティリティ・データベースの使用」

次の dbping コマンド

```
dbping -c "uid=dba;pwd=sql;eng=servername;dbn=databasename" -d
```

は、servername というサーバ上で実行されている databasename というデータベースを検出します。これに相当する ODBC のソースは、次のようになります。

```

HENV henv;
HDBC hdbc;
char * con_str = "uid=dba;pwd=sql;eng=servername;dbn=databasename;Driver={Adaptive
Server Anywhere 7.0}";
bool ok;

ok = SQLAllocEnv( &henv );
if( ok ) {
ok = SQLAllocConnect( henv, &hdbc );

```

```
if( ok ) {  
    success = SQLDriverConnect( hdbc, (SQLHWND)NULL, (SQLCHAR *)con_str, SQL_NTS, NULL, 0,  
    NULL, SQL_DRIVER_NOPROMPT );  
    ...  
    if( success ) {  
        SQLDisconnect();  
    }  
}  
SQLFreeConnect( hdbc );  
}  
SQLFreeEnv( henv );
```

#### 上記の例についての注意

1. 上記の例では、任意の有効な接続文字列を指定できます。接続文字列が dbping で使用可能な接続文字列であれば、それに相当するアプリケーションの検出用モジュールでも使用できます。必要に応じて、接続文字列パラメータを追加してください。
2. 上記の例では、エラー処理コードは省略されています。