

【T3】Delphi・C++テクニカルセッション



DEVELOPER CAMP

「Delphi/C++Builder DB総ざらい」

株式会社日本情報システム
筑木 真志

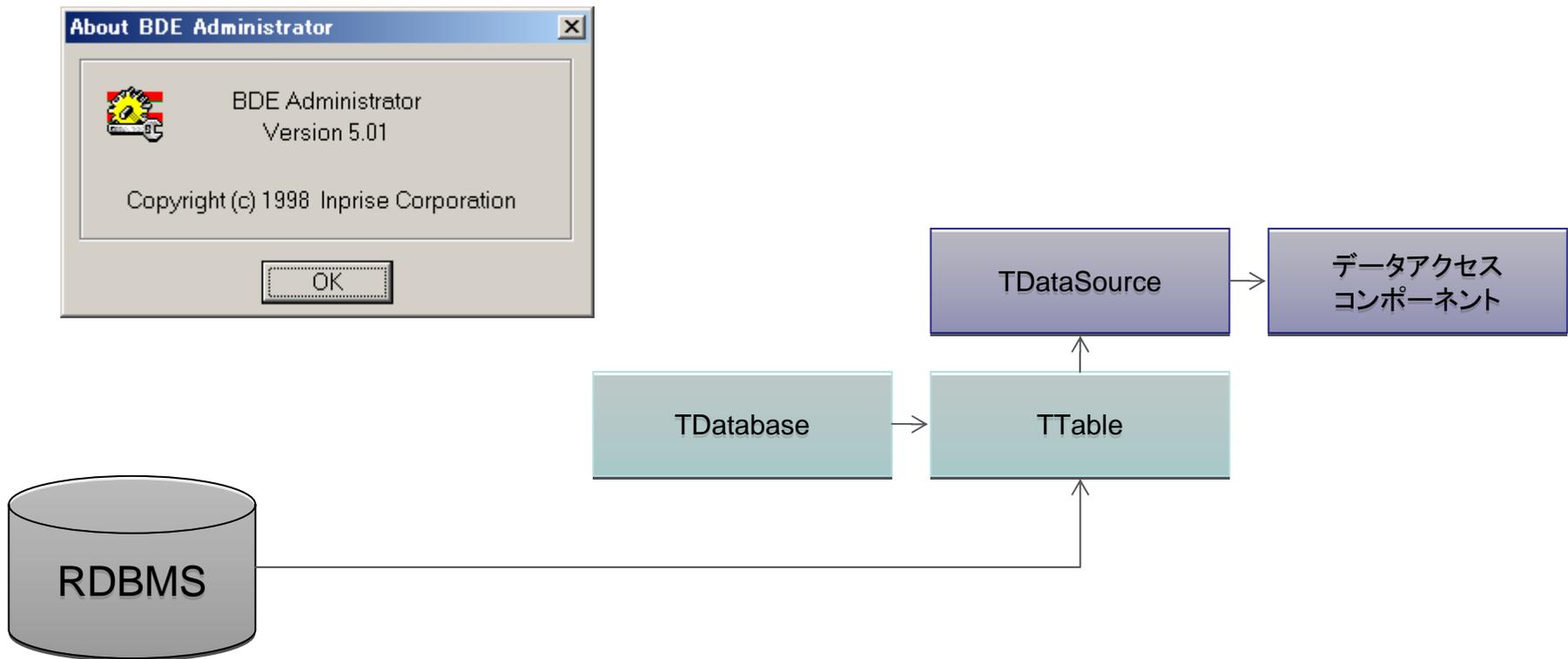
- Delphi / C++Builderからデータベースへ接続する際の手順についての概要をまとめて解説します
 - Delphi / C++Builderデータベースプログラミングフレームワーク
 - Firebird / InterBase
 - MySQL
 - PostgreSQL
 - Oracle / SQL Server / DB2
 - C++Builderで埋め込みSQLを使用する(Oracle / DB2)



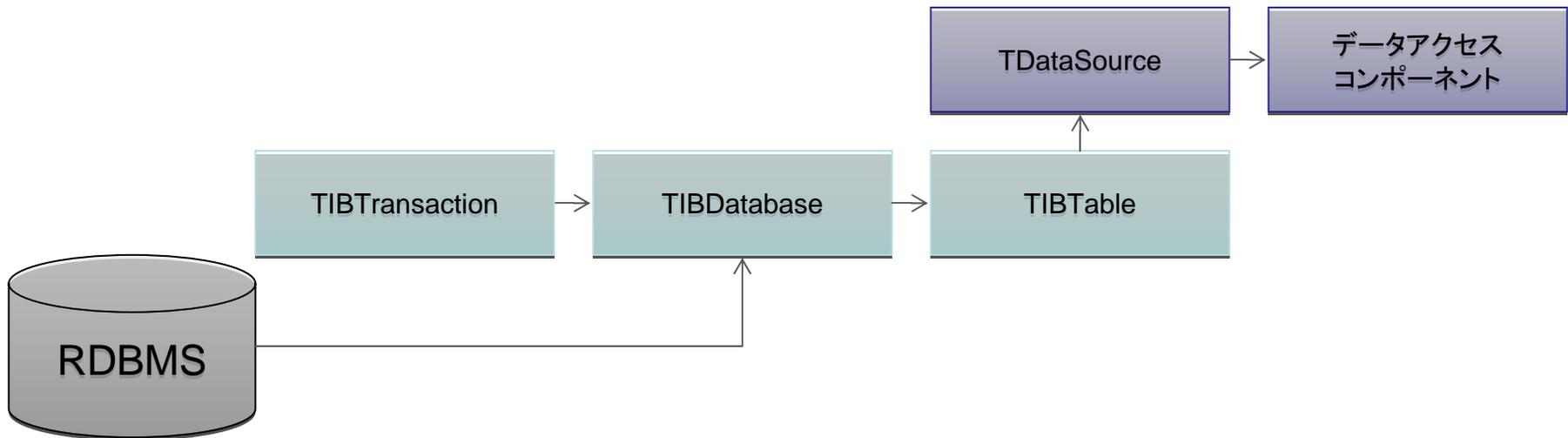
Delphi/C++Builderデータベースプログラミングフレームワーク

- Delphi/C++Builderは標準で以下のデータベースプログラミングフレームワークをサポートしています
 - BDE (Borland Database Engine)
 - InterBaseExpress
 - dbGo (ADO/OLEDB)
 - dbExpress
- その他サードパーティ製品、オープンソース製品も多数
 - ZeosLib <http://zeos.firmos.at/>
 - FIBPlus <http://www.devrace.com/en/fibplus/>
 - IBOjects <http://www.ibobjects.com/>
 - UniDAC <http://www.devart.com/dac.html>

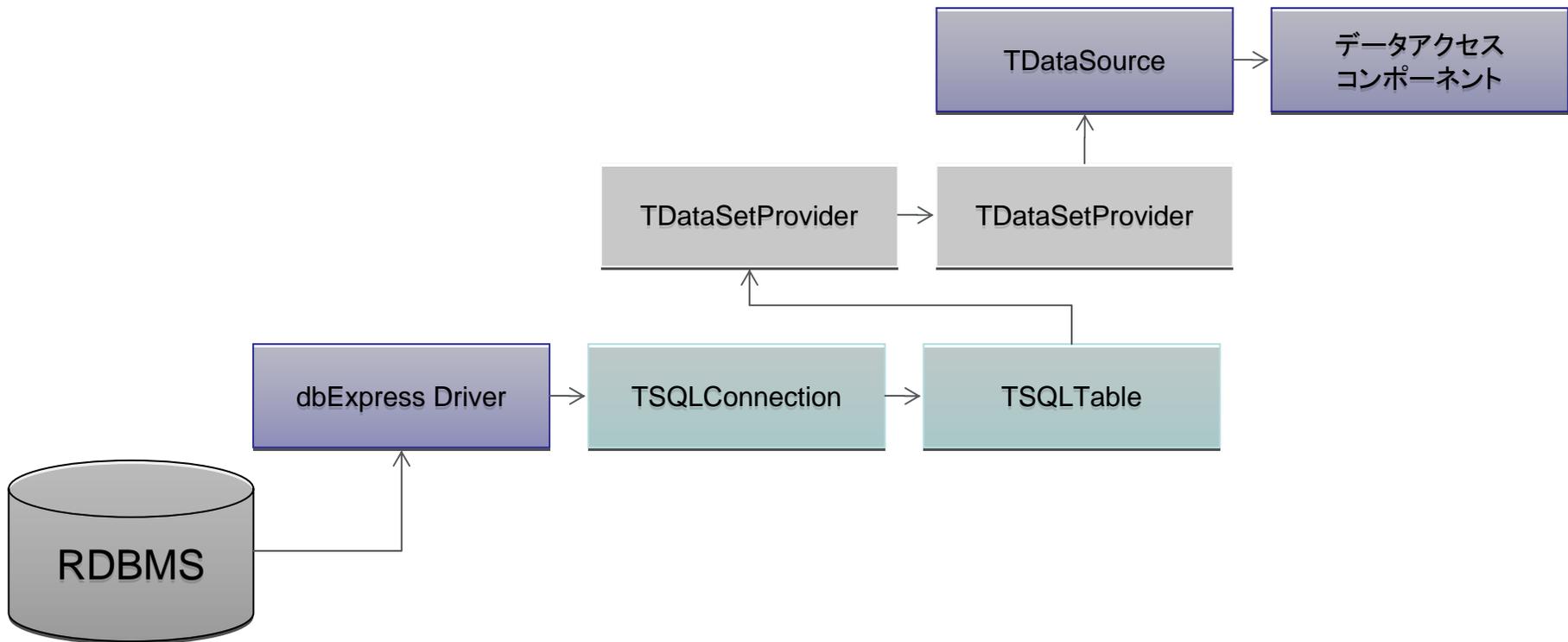
- BDE は、Paradoxをベースとしたフレームワーク。異種DB間でのクエリーのサポートなど、高度な機能もあるが、すでに開発は中止。とはいえ、結構使われている



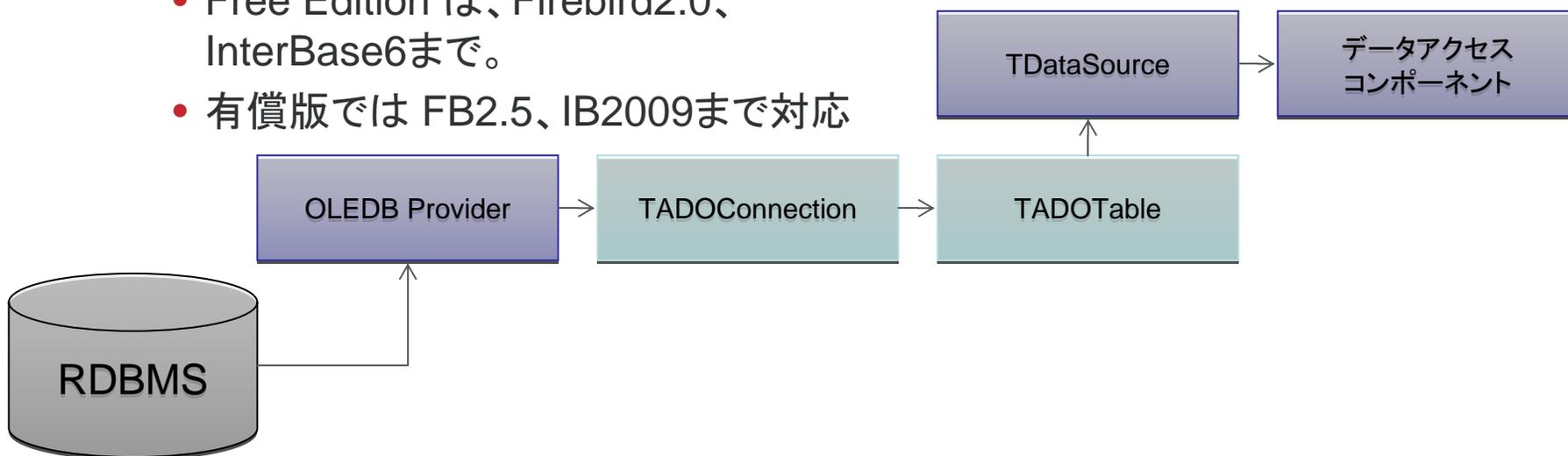
- InterBaseExpress は、InterBaseと接続するためのフレームワーク。InterBaseAPIを直接利用するコンポーネントで構成されている。ベースは、98年に公開されたFreeIBComponents。
 - TIBTransactionコンポーネントが分離しているのが特徴
 - 複数のデータベースへの二相コミットに対応出来る



- dbExpress4は、Delphi6から導入されたdbExpressの最新版。Delphi2007でversion 4.0になったが、これまでのドライバも使うことは出来る
- 2009のProfessional版はEULAによりリモート接続は不可



- dbGo は、ADO (ActiveX Data Object) をベースとしたフレームワーク。Delphi5でADOExpressとして導入されて、Delphi6からdbGoという名前になった
 - Professional版でもOracleやMSSQLなどに接続できる
 - Firebird / InterBaseに接続するためには、ibProvider が利用できる
 - ibProvider
 - <http://www.ibprovider.com/eng/>
 - Free Edition は、Firebird2.0、InterBase6まで。
 - 有償版では FB2.5、IB2009まで対応



- ZeosLib は、Delphi/C++Builder/Kylix/Lazarus に対応したオープンソースのライブラリ
 - <http://zeos.firmos.at/>
 - 現在の最新版は6.6.5、Delphi2007まで対応
 - Delphi2009対応は、今後出る予定のver.7までお預け
 - 対応するデータベースが多いのが特徴
 - MySQL (normal and embedded)
 - PostgreSQL
 - Firebird (normal and embedded)
 - Interbase
 - SQLite
 - MS SQL Server
 - Oracle
 - Sybase
 - ADO Connections

ver.7での対応予定

- Delphi7～Delphi2009サポート
- FPC/Lazarusサポート
- MSEide/MSEguiサポート

- ADO
- MySQL 4.1～6
- InterBase 5 / 6
- Firebird 1.0～2.1
- PostgreSQL 7～8
- SQLite 2.8～3

- Preparedステートメントのサポート
- Firebird/InterBase以外のDBでイベントサポート
- プロパティエディタの改良

- Delphiから各種データベースへの接続を行う際には、通常は各データベース毎のクライアントライブラリが必要だが、直接データベースと対話出来るプロトコルを実装したものをネイティブコンポーネント(またはネイティブドライバ)と呼ぶ。IBXもその一例。
 - Direct Oracle Access 最新版は4.1.2.0
 - Delphi2009 対応、Oracle11g対応
 - <http://www.allroundautomations.com/doa.html>
 - <http://www.teppi.com/Components/DOA/>
 - VCL Data Access Components (ODAC, MyDAC, PgDAC)
 - Delphi2009対応、Oracle11g / MySQL6.0 / PostgreSQL8.3 に対応
 - <http://www.devart.com/dac.html>



EMBARCADERO
TECHNOLOGIES®

DEVELOPER CAMP

Firebird / InterBase

- Firebird / InterBase は、95年にDelphi1 C/SにInterBase4.0Cがバンドルされて以来、Delphiには必ずInterBaseのローカル版や開発版がバンドルされてきた。
- 2000年にInterBase6がオープンソース化された後、Firebirdがオープンソースプロジェクトとしてフォーク。以降、InterBaseは、6.5 / 7 / 7.1 / 7.5.1 / 2007 / 2009 とバージョンを重ねている。FirebirdもInterBase6と互換性のある1.0から、1.5 / 2.0 / 2.1とバージョンを重ね、両者の違いも大きくなってきているが、今のところAPIレベルの互換性はほぼ保たれている。
- FirebirdのオープンソースライセンスはMPLベースのIPL/IDPL。

- Firebird / InterBaseへの接続方法は多種多様なので、ここではdbExpressを基本として説明します。
 - dbExpress
 - 標準INTERBASEドライバ (Firebirdはサポート外)
 - 標準Firebirdドライバ (Delphi/C++Builder 2010よりサポート)
 - Devart dbExpress driver for InterBase/Firebird
 - Upscene InterXpress for Firebird
 - dbExpress Driver for Firebird
 - IBX
 - FIBPlus
 - ADO/ODBC経由での接続
 - ibOleDb
 - ZeosLib

- dbExpressでFirebirdに接続する。
 - テスト条件: Firebird 2.1.2、DBCharSet=UTF8
 - 標準ドライバでパラメータクエリは注意が必要
 - Devartドライバ
 - 規定値ではUnicode対応がDisableになっているので、TSQLConnectionのParamsに、UseUnicode=Trueを追加する必要がある。
 - INTERBASEドライバ
 - FirebirdのUTF8データベースへの接続には、ServerCharSet=UTF8を指定
InterBase2009の付属ドキュメントではUTF-8とあるが、「-」は使えないのでこれは間違い。
 - UTF8を指定して接続した場合、文字型のパラメータがftWideStringになる。この時にAsStringで文字列を渡すとエラーになるので、AsWideStringを使う。
 - InterXpressドライバ
 - 現状ではUnicode対応が出来ていない
 - その他にも動作に注意が必要

- dbExpressでInterBaseに接続する。
 - テスト条件: InterBase2009、DBCharSet=UTF8
 - Firebirdと同様、標準ドライバでパラメータクエリで注意が必要
 - Devartドライバ
 - 規定値ではUnicode対応がDisableになっているので、TSQLConnectionのParamsに、UseUnicode=Trueを追加する必要がある。
 - INTERBASEドライバ
 - UTF8データベースへの接続なので、ServerCharSet=UTF8を指定する
 - UTF8を指定して接続した場合、文字型のパラメータがftWideStringになる。この時にAsStringで文字列を渡すとエラーになるので、AsWideStringを使う。
 - uses節に、DBXInterBase を追加すること。これがないと、メタデータを取得できないというエラーが発生する。(GetMetadata=Falseでも・・・)

- dbExpressのデータをDB系コンポーネントで利用する。(共通)
 - 一番簡単なのは、TSimpleDataSetを利用すること
 - TSimpleDataSetは、TSQLConnection+TSQLDataSet+TClientDataSetで一つのコンポーネントにしたようなもの。
 - Connectionだけ別のコンポーネントを指定することも可能なので、複数のTSimpleDataSetで接続を共有することも可能。
 - TSQLDataSet+TDataSetProvider+TClientDataSetと接続して使用した場合と比べると、Providerの挙動を指定出来ないの細かいことには向かない。
 - dbExpressでデータベース系コンポーネントを使うなら、これを使うと大変楽が出来る。
 - TSQLDataSet / TSQLQuery / TSQLTable等をTDataSetProvider+TClientDataSetと接続して利用する
 - Insert / Update / Delete は、プロバイダがSQLを自動生成してくれる。
 - 既定値のままだと、Whereに全ての項目を指定したUpdate / Delete動作になるのでこれは変更する。
 - 元になるデータセットのフィールドコンポーネントの ProviderFlagsで、キー項目(普通は主キー)のpflnKeyをTrueにする。
 - TDataSetProviderのUpdateModeをupWhereKeyOnlyにする。
 - Firebird/InterBaseの場合、SQLを大文字で書くこと。

- blobの扱い

- BLOB SUB_TYPE 1 (TEXT)のBlobは、WideMemo型のフィールドにマッピングされる。TField.OnGetTextイベントでそのまま展開可能。

- Text := Sender.AsString;

- その他の画像データなどのBLOBのINSERT

- パラメータに対して、Assign(hoge) とすればよい
- 標準ドライバではFirebirdにBlobをInsert出来ない
- InterBase2009にはもちろん出来る



EMBARCADERO
TECHNOLOGIES®

DEVELOPER CAMP

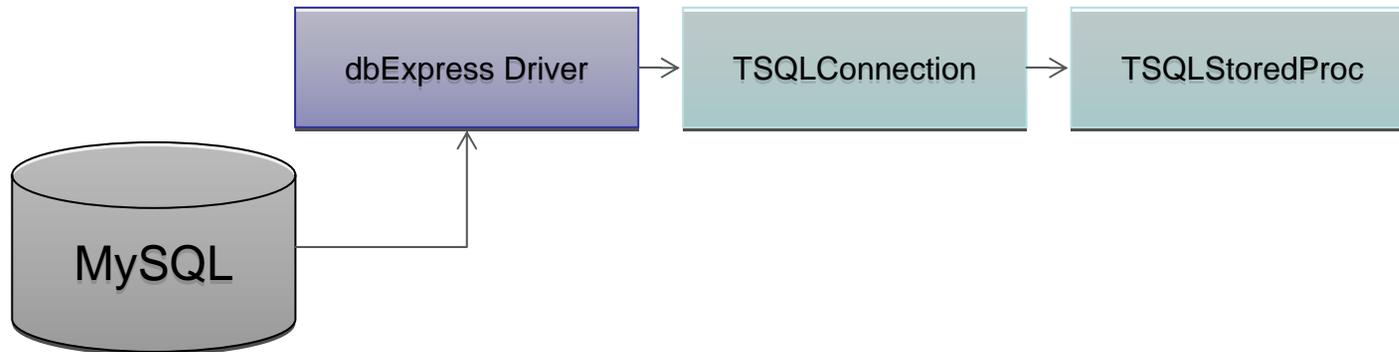
MySQL

- MySQLは、「世界でもっとも普及しているオープンソースデータベース」で、開発元のMySQL社がサンマイクロシステムズ社に買収され、今回Oracleによって買収されました。
- 最新安定版は最新安定版は5.1.37(13/7/2009) で、6.0-Beta(6/4/2009)がリリースされて開発中だったが、5.4-Beta(17/6/2009)が突然リリースされています。これは、5.1のパフォーマンス改善版であるとのことだが、現状では5.1より遅くなるケースもあるとのこと。
- MySQLはストレージエンジンを差し替えて動作することが出来るのが特徴だが、6.0ではInterBaseの生みの親のJimStarkeyになるFalconストレージエンジンが実装されている。FalconはMVCCによる完全なトランザクション制御とメモリ上でのバージョンニングによる高速なロールバックを可能としている。
- オープンソース版と有償版があり、オープンソース版のライセンスはGPL。

- MySQLへの接続方法はdbExpress、ダイレクト接続、ADO/ODBCなどなどがある。
 - dbExpress
 - MySQLドライバ
 - Devart dbExpress driver for MySQL
 - Devart MyDAC --- direct / non direct access
 - ADO/ODBC経由での接続
 - ZeosLib

- dbExpressでMySQLに接続する。
 - テスト条件: MySQL 5.0.83、ServerCharSet=utf8
 - 標準ドライバ
 - uses節に、DBXMySQL を追加すること。これがないと、メタデータを取得できないというエラーが発生する。(GetMetadata=Falseでも…)
 - 標準ドライバのサポートは4.1と5.0のみ。
 - MySQL6.0からサロゲートペアがサポートされる予定だが、6.0自体がキャンセルになってしまった…。
 - その他は、特に問題無し。
 - Devart MySQLドライバ

- dbExpressからストアプロシージャを使用する。
- MySQL5.0からストアプロシージャがサポートされている。
- TSQLStoredProcのSQLConnectionを設定し、必要なら入力パラメータ (Paramtype=ptInput) のパラメータを設定して、ExecProcするだけ。





EMBARCADERO
TECHNOLOGIES®

DEVELOPER CAMP

PostgreSQL

- PostgreSQLは、Ingresから派生したオープンソースのORDBMS(自称)です。日本ではSRA OSS社や富士通が担っています。PostgreSQLはFirebirdと同様、バージョンニング・アーキテクチャに基づくRDBMSです。
- 最新安定版は8.3.7(16/3/2009)です。8.3ではHOP(Heap Only Tuples)という新機能が追加され、更新処理が劇的に改善されています。レプリケーションやクラスタへの対応等でも、対応製品が多くOracleの対抗馬と位置付けられます。
- 属性として幾何データを持たせられる。PostGISによる拡張。
- PostgreSQLのライセンスはBSDライセンス。

- PostgreSQLへの接続方法はdbExpress、ダイレクト接続、ADO/ODBCなどなどがある。
 - dbExpress
 - Devart dbExpress driver for PostgreSQL --- DirectAccess
<http://www.devart.com/dbx/>
 - PgExpress driver --- Needs client library
<http://www.vitavoom.com/products/pgedriver/>
 - Devart PgDAC --- direct access
 - ADO/ODBC経由での接続
 - ZeosLib

- dbExpressでPostgreSQLに接続する。
 - テスト条件: PostgreSQL 8.3.7、Encoding=UTF8
 - 標準ドライバはないので、Devartドライバ又はpgExpressドライバが必要だがpgExpressはDelphi2009に正式対応していない上、Devartドライバはクライアントライブラリ不要なので、Devartがお勧め
 - Devartドライバ
 - 規定値ではUnicode対応がDisableになっているので、TSQLConnectionのParamsに、UseUnicode=Trueを追加する必要がある。



dbGoを使った商用データベースへの接続

- dbGoを使用すれば、Professional版ではサポートしていない、RDBにも接続可能
 - Microsoft SQL Server 2000/2005/2008
 - Microsoft Access 2003 (Microsoft JET 3.x/4.x)
 - Microsoft Access 2007 (Microsoft Office 12.0 Access Database Engine)
 - Oracle 8i/9i/10g/11g
 - IBM DB2 8.x/9.x
 - など
- 基本的な使用方法はDBExpressと変わらない
 - TSQLConnection → TADOCConnection
 - TSQLTable → TADOTable

- Oracleの場合
 - Oracle Provider for OLE DB (Oracle製)
 - Microsoft OLE DB Provider for Oracle (Microsoft製)
- SQL Serverの場合
 - SQL Server 2008 Native Client
 - Microsoft OLE DB Provider for SQL Server

- 標準のクライアントから普通に接続出来るか？
 - SQL Plus / SQL Developer (Oracle)
 - SQL Server Management Studio (SQL Server)
 - コントロール・センター (DB2)
 - など
- TADOTableのTableNameプロパティにスキーマ名を含める
例) スキーマ名: Foo テーブル名: Bar
TableName = "Bar"; → TableName = "Foo.Bar";
- TADOConnectionのCursorLocationプロパティの値を変更する
 - clUseServer(カーソルをサーバー側で持つ)に変更する



C++Builderで埋め込みSQLを使用する

- C/C++のソースコードに直接SQL文を記述して実行する

```
int main(int argc, char* argv[])
{
    EXEC SQL BEGIN DECLARE SECTION;
    VARCHAR filename[8];
    VARCHAR mapname[128];
    double frame_lu_b;
    EXEC SQL END DECLARE SECTION;

    EXEC SQL CONNECT :user IDENTIFIED BY :passwd USING :host;

    EXEC SQL DECLARE CURSOR1 CURSOR FOR
        SELECT FILENAME, MAPNAME, FRAME_LU_B FROM MAPTABLE ORDER BY FILENAME;

    EXEC SQL OPEN CURSOR1;
    EXEC SQL WHENEVER NOT FOUND DO BREAK;

    while (1) {
        EXEC SQL FETCH CURSOR1 INTO :filename, :mapname, :frame_lu_b;
        printf("FILENAME = %s, MAPNAME = %s frame_lu_b = %.4f¥n",
            filename.arr, mapname.arr, frame_lu_b);
    }

    EXEC SQL CLOSE CURSOR1;

    return 0;
}
```

- 特徴

- データベースとのやりとりはソースコードに直接記述する
- プリコンパイラでホスト言語(C/C++)に変換してからコンパイルする
- データベースのAPIを直接呼び出すのでパフォーマンスがよい
- クエリ結果を配列にまとめてフェッチできる

- 欠点

- デバッグが非常に面倒
- データベースの実装に大きく依存するため汎用性が悪くなる
- 文字コードの変換等は自前で行う必要がある
- GUIとのやりとりも自前で行う必要がある

● ホスト変数

- ホスト言語と埋め込みSQLとの間でデータをやりとるする変数
- C/C++の基本型のみ可 (Oracleの場合、VARCHAR型が用意)
- ホスト変数は、BEGIN DECLARE SECTIONとEND DECLARE SECTIONでくる

```
EXEC SQL BEGIN DECLARE SECTION;  
  VARCHAR filename[8];  
  VARCHAR mapname[128];  
  double frame_lu_b;  
EXEC SQL END DECLARE SECTION;
```

- ホスト変数をSQLに組み込むには、変数の前にコロン(:)をつける

```
EXEC SQL FETCH CURSOR1 INTO :filename, :mapname, :frame_lu_b;
```

● 実行結果

- 変数sqlca.sqlcodeでSQLの実行結果(エラーコード)を参照する

- `coff2omf.exe`を使って、
`$(ORACLEHOME)¥precomp¥LIB¥orasql11.lib`を
C++Builderで使用できるようにする

```
coff2omf orasql11.lib orasql11omf.lib
```

- プリコンパイラの実行

```
proc code=cpp cpp_suffix=cpp File1.pc
```

- implib.exeを使って、\$(DB2PATH)\bin\db2app.dllのC++Builder用のインポートライブラリを作る

```
implib db2app.dll db2appOMF.lib
```

- プリコンパイラの実行

- DB2のコマンド・ウィンドウで以下を実行する

```
db2 connect to db2 user scott using tiger
db2 prep DB2Test.sqx OUTPUT DB2Test.cpp
db2 connect reset
```

- インクルードパスの設定

- \$(DB2PATH)\includeを検索パスの最初にする

ご静聴ありがとうございました!!

Special thanks to:

アナハイムテクノロジー株式会社 はやし つとむ様