

【B2】Delphi/C++Builderテクニカルセッション

BDEアプリを最新へ FireDACによる移行

エンバカデロ・テクノロジーズ
エヴァンジェリスト 高橋 智宏

BDEアプリを最新へ – FireDACによる移行

はじめに

はじめに

- ◎ BDEの現状
- ◎ BDEから移行がなぜ必要なのか
- ◎ BDEからの移行の壁
- ◎ FireDACについて
- ◎ Paradox/dBaseユーザー向けの移行
- ◎ BDEからFireDACへ移行のポイント
- ◎ FireDACのデモ
- ◎ まとめ

BDEアプリを最新へ – FireDACによる移行

BDEの現状

BDEとは？

- 旧Borland社が提供したデータベースアクセスを共通化するミドルウェア(古くはIDAPI と呼ばれたDBアクセスエンジン)
- 複数のアプリケーションで共有できるデータアクセスメカニズム
- BDEと併せて“**SQL-Link**”というミドルウェアを利用することで主要なRDBMSへネイティブ接続が可能
- Delphi/C++Builderで開発されたアプリケーションと共に利用され、広く普及している

BDEの特徴 (その1)

◎ 共通の手順で様々なDBへアクセス可能

BDEユーザーはデータベースの違いを意識することなく、異なるシステムへ同じ手順でアクセスできる

◎ BDEの機能を直接使用するための API を提供

BDE API は、Windows DLL をロードし、その中の関数を使用できるプログラミング言語であれば、どの言語からも呼び出せる(Delphi/C++Builder以外からも利用可)
BDEオブジェクトは、BDE API(DBIxxxx)をラップしたコンポーネント

◎ 幅広いデータベースのサポート

Paradox、dBASE、AccessなどのファイルベースDB

InterBase、Oracle、Sybase、MSSQL、DB2、Informix などのRDBMS

ODBCを介してさらに多くのデータベースへ接続も可能

BDEの特徴 (その2)

◎ 異種結合

異なるデータベースに属する 2 つのテーブルを結合可能

例えば、

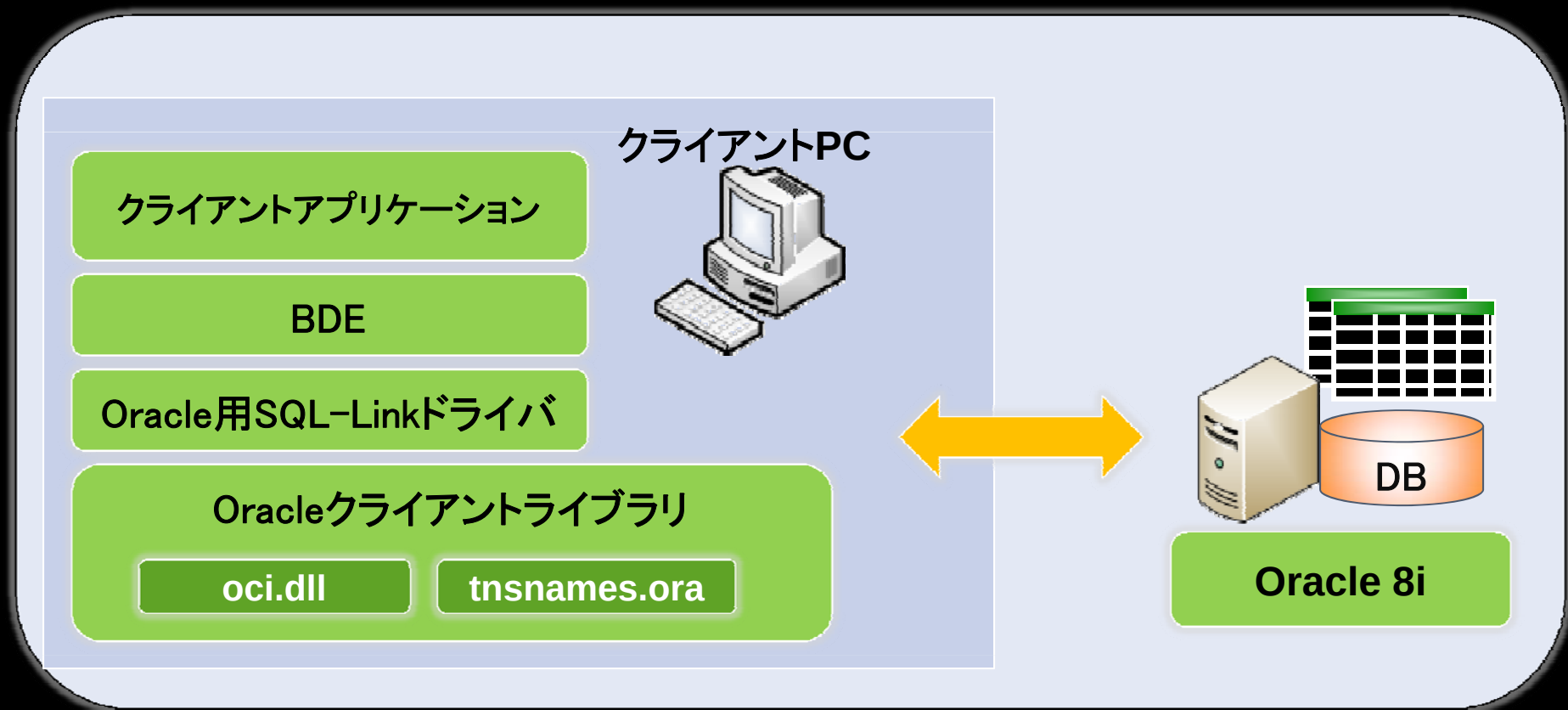
```
SELECT *  
FROM ":DBDEMOS:Customer.db" C, ":BCDEMOS:Orders.db" O  
WHERE (C.CustNo = O.CustNo)
```

異種結合できるのは BDE(ローカル, ODBC, または SQL-Link)で アクセス可能なデータベースタイプのテーブル

BDEのよくある旧システムの構成例

Oracle 8i をターゲットに Delphi 7で作成されていたC/Sシステム

(Delphi 7 + BDE + SQL-Link for ORACLE + Oracle 8i)



BDEの現状

◎ BDEは2002年に開発・保守終了

不具合があっても修正パッチの提供は無し

◎ BDEの最新バージョンは、5.2

- ・Delphi 7/C++Builder 6以降、BDEのバージョンは更新されていない
- ・現在のDelphi /C++Builder製品に付属するBDEは、あくまで過去の資産の保守用

◎ 動作保証プラットフォームは、Windows XPまで

Windows XP以降にリリースされたOSバージョンは、動作保証外
(例えば、Windows Vista/7、Windows 2003/2008 Serverなど)

BDEアプリを最新へ – FireDACによる移行

BDEからの移行がなぜ必要なのか

BDE+Paradoxについて

- ◎ 現在のBDEは、Paradox/dBaseの保守用
- ◎ BDE + Paradoxの短所 (RDBMSと比較して)
 - ・トランザクションや排他処理が貧弱
 - ・テーブルのデータが破損しやすい(そのリスクが大きい)
 - ・Paradoxのテーブルサイズの制限(ブロックサイズ × 65535)
 - ・データ量に比例し、パフォーマンス低下やエラーが発生しやすい
 - ・Unicodeは当然扱えない

ますますデータの肥大化が進む現状のシステムにおいて
これらの問題を抱えたままParadoxを使い続ける意味があるのか？

BDEの抱える問題 (その1)

◎ 近年の(PC)ハードウェア構成にマッチしない

- ・4GB越えのHDDの空き容量が正しく認識しない
- ・マルチコア/プロセッサ搭載PC上での動作不安

◎ Vista以降のOS機能に未対応

- ・UAC(User Account Control)
- ・ASLR(Address Space Layout Randomization)によるアドレス衝突が発生
- ・BDEアプリを複数同時起動した場合、初期化エラーが発生することがある

PCやWindows、そしてBDEの設定を色々変更すれば
なんとか動作するが、一切の保証は無し

BDEの抱える問題 (その2)

◎ 対応しているデータベースのバージョンが古い

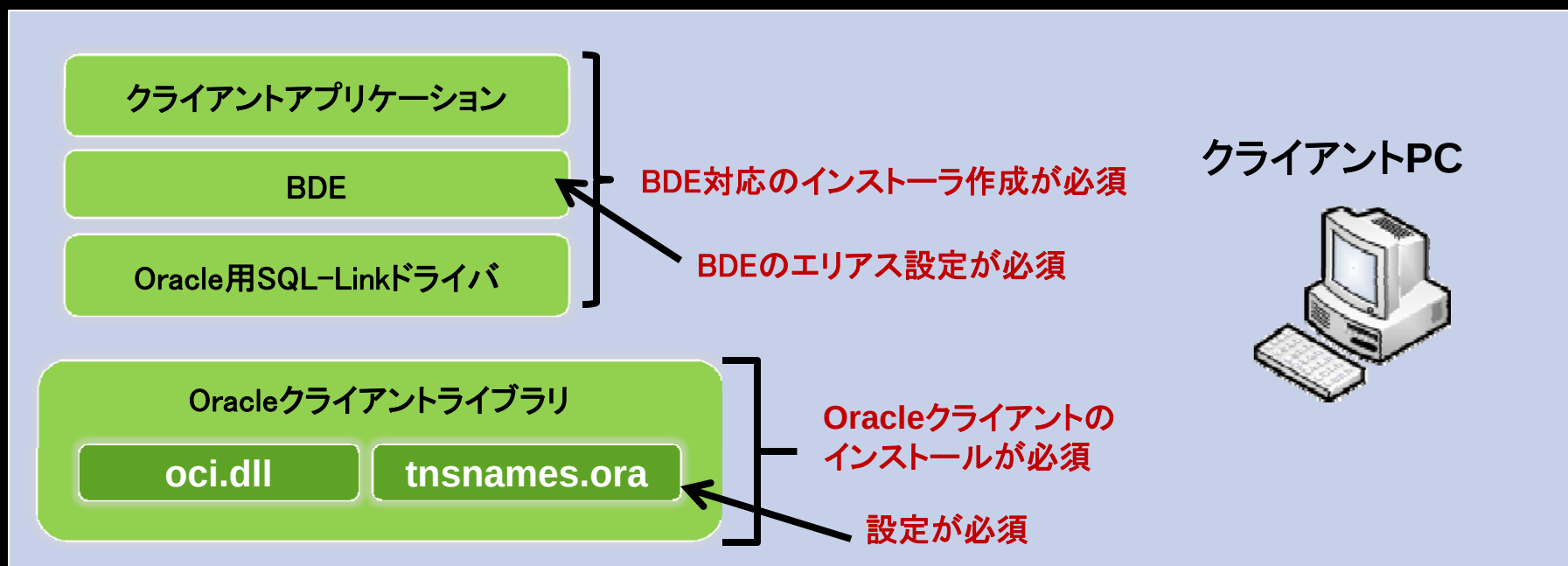
対応RDBMS:	InterBase	5 for Linux 5.6 for Linux 5.5 for Windows 95/NT 5.6 for Windows 95/NT 6 for Windows 6 for Linux	InterBase 5.5J Client InterBase 5.6J Client InterBase 5.5J Client InterBase 5.6J Client InterBase 6 Client InterBase 6 Client
	MS SQL SERVER	6.50(SP2) 6.50(SP3) 6.50(SP4) 7.0	DB-Library 6.50.240 DB-Library 6.50.258 DB-Library 6.50.281 DB-Library 7.00.623
	ORACLE	8.1.5 8.1.6	Net*8.1.5 Net*8.1.6
	SYBASE	SYSTEM 11.0.3	OPEN Client 10.0.3 OPEN Client 10.0.4
		Adaptive Server 11.5	OPEN Client 11.5
	INFORMIX	DS 7.2 Universal 9.11	INFORMIX CONNECT 7.2 Developer SDK9.12.TC2
	DB2 DB2	2.1.1 5.0	2.1.1 CAE 5.0 CAE

ODBC接続によって一応新しいバージョンへ繋がるが
一切の保証は無し

BDEの抱える問題 (その3)

◎ とにかく配布が面倒

例えば、BDE + SQL-Link for ORACLE + Oracle 8iの構成の場合



異なるBDEのバージョン(例えば、4.x と 5.x)の混在や、エリアスの重複など
実行環境に依存し、問題が生じるケースも多い

BDEの抱える問題 (その4)

◎ 最近のニーズに適したアプリケーションが作成できない

- ・64ビットアプリ化
- ・Unicodeへ対応
- ・クロスプラットフォーム化
- ・マルチデバイス
- ・多層データベースアプリケーションへの対応

など

BDEを使い続けることのリスク

◎ メーカーからのサポートが期待できない

既にサポートが終了しているため、もしシステムに深刻な問題が発生しても
自己責任で利用、回避が必要

◎ BDEは不安定な土壌で動作しているという認識が必要

- ・あくまで旧システムとの互換性の範囲で動作しているに過ぎない
- ・今後普及していくデバイス、Windows OS、データベース等でも同じ仕様で利用できるとは限らないため、もし仕様が変われば、いずれ動作しなくなる

◎ 既存のアプリケーション資産を様々な分野へ展開できる可能性を自ら閉ざしている

- ・実行環境が固定されるため、アプリケーションの更なる発展は見込めない

”その場しのぎ”や、”延命”も
そろそろ限界じゃありませんか？

既存のBDEアプリを最新にアップデートしましょう！

では、それをいつやるのか？

今でしょ！

BDEアプリを最新へ – FireDACによる移行

BDEからの移行の壁

BDEから移行するための選択肢

BDE以外で、従来のDelphi/C++Builderから
標準で利用可能なデータベースフレームワークの種類

- ◎ IBExpress (InterBase専用)

- ◎ dbGO(ADO/OLEDB)

- ◎ dbExpress

※サードパーティ製品は除く

おそらくこれらのコンポーネントを利用し、BDEから
移行を一度は検討された方も多いと思います

サポートへよくあるお問い合わせ(1)

- 現在、BDEのTTable(TQuery)から移行中です。
ところがdbExpressのTSQLTable(Query)では、
例えば、以下に該当するメソッドやプロパティの見つかりません。

TTable.FetchAll

TQuery.Constrained

TTable.UpdateObject

TQuery.Data

TQuery.RequestLive

TTable.exclusive

存在しないプロパティやメソッドの代替案を教えてください。
上記に依存しているコードが多く、できれば修正したくありません。

BDE DataSetとの非互換性の例

サポートへよくあるお問い合わせ(2)

- ◎ Oracleテーブル(レコード数は約100万件)のオープンにかかる時間は
BDE → 1秒
dbGO(ADO)やdbExpress → 21秒 かかりました。

テーブルのアクセス方法は、

Tableコンポーネントでは、特にフィルタは設定してません。

Queryコンポーネントでは、”select * from テーブル名”で結果セットを取得しており、where句によってフィルタはしていません。

なぜ同じコードで、BDEとこんなにパフォーマンスが違うんですか？

データアクセスアーキテクチャの違いにより
実行パフォーマンスが出ない典型的な例

BDEからの移行をより困難にする理由

- ◎ データアクセスアーキテクチャの違い

- ・サーバーサイドカーソルか、クライアントサイドカーソルか
- ・フィルタの指定有無/条件など

- ◎ BDE DataSetと非互換(メソッドやプロパティの違いなど)

- ◎ 同等の実行パフォーマンスが出ない

上記のような理由により、コードの修正が発生し、
それにかかる工数が大きいため、なかなか移行に踏み切れない

BDEから移行する新たな選択肢

BDE以外で、Delphi/C++Builderから利用可能な
データベースフレームワークの新定番

- ◎ IBExpress (InterBase専用)
- ◎ dbGO(ADO/OLEDB)
- ◎ dbExpress
- ◎ FireDAC **New!**

※サードパーティ製品は除く

BDEアプリを最新へ – FireDACによる移行

FireDACについて

FireDACとは？



- 広範なデータアクセスコンポーネント
 - あらゆるデータベースアプリケーション開発向け
 - Delphi および C++Builder 向け
 - Win32, Win64, Mac OS X
 - Delphi 5～XE3, C++Builder 6～XE3
- ハイパフォーマンス且つ、簡単にエンタープライズデータベースに接続可能
- 統一的なデータアクセスが可能
 - しかも、データベースに特有な機能も利用可能

FireDACのネイティブドライバ



Unicodeに対応!!

- MySQL 3.21以降
- Microsoft SQL Server 2000以降
- Microsoft SQL Azure
- Oracle 8.0.3以降
- InterBase 6以降
- PostgreSQL 7.4以降
- DataSnap
- SQLite 3.x
- Sybase SQL Anywhere 5.0以降
- Microsoft Access 2000以降
- IBM DB2 8.1以降
- Firebird 1.5以降
- Advantage Database 8.0以降
- ODBC接続
- dbExpressブリッジ接続

FireDACの構造



FireDACを使用するメリット

◎ BDEとの互換性

- ・データアクセスアーキテクチャが類似
- ・従来のBDE Datasetとの互換性を備えたDataSetクラス
- ・CachedUpdatesモードも利用可能

◎ 配布がとても簡単

- ・専用のデータベースドライバやインストーラーが不要

◎ BDEと同等以上のパフォーマンス

◎ Professionalエディションでもリモート接続が可能

BDEアプリを最新へ – FireDACによる移行

Paradox/dBaseユーザー向けの移行

データベースの移行について

◎ FireDACは、Paradox/dBaseを未サポート
Paradox/dBaseから別のデータベースへ移行が必要

◎ 代替データベースに関する指針

- 各種 RDBMSへ移行
- 同じタイプのローカルDBへ移行

例えば、

- XMLファイル
- MS Access
- SQLite
- InterBase Desktop Editionなど

テーブルのデータを移行するには

◎ BDEを介し、DataPump(データポンプ)を利用する

- ・データベースへの接続は、SQL-LinkもしくはODBCで行なう

注意点:

- ・BDEのエリアスを予め設定しておく
- ・移行先の(入れ物となる)データベースファイルを予め作成しておく
- ・このユーティリティが付属しているバージョンは、C++Builder6 ,Delphi7 迄

◎ Paradoxから各種RDBMSへ移行するサードパーティ製のツールを利用

◎ 自分でデータ移行用のプログラムを作成する

◎ TClientDataSetを介してXMLファイルへ出力

- <http://blogs.embarcadero.com/teamj/2013/03/06/3722/>

BDEアプリを最新へ – FireDACによる移行

BDEからFireDACへ移行のポイント

BDEとFireDACのコンポーネント名

BDE

- TDatabase
- TSession
- TTable
- TQuery
- TStoredProc
- TUpdateSQL
- TBatchMove

など

FireDAC

- TADConnection
- TADManager
- TADTable
- TADQuery
- TADStoredProc
- TADUpdateSQL
- TADDataMove

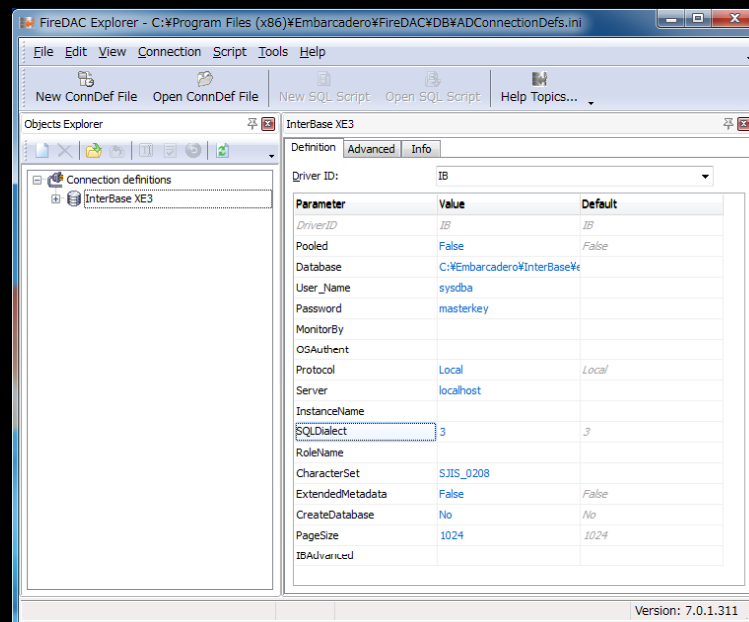
※ 一部対応するコンポーネントがあっても、挙動が異なるものがあります

FireDACのコンフィグファイル

- BDE Administratorと同様に、FireDAC Explorerによってエリアスの情報を設定ファイルに保存できる
 - ・BDEでは、IDAPI.CFGに保存
 - ・FireDACでは、ADConnectionDefs.iniに保存

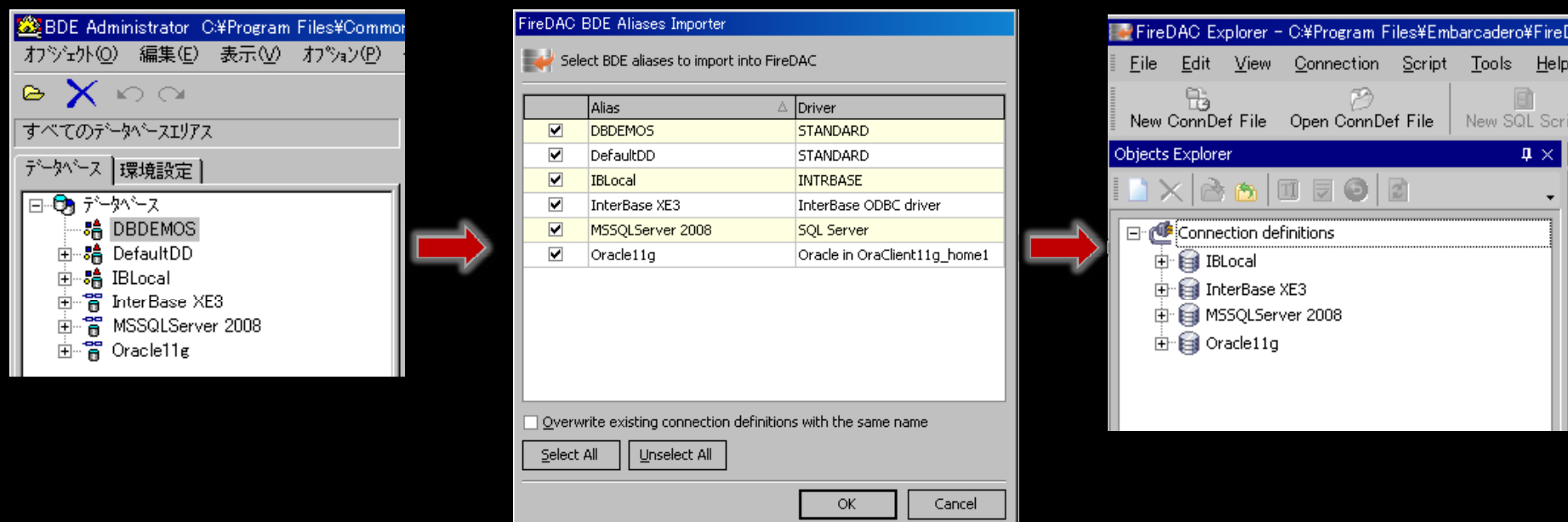
注意:

Windows Vista以降のOSで
デフォルトのADConnectionDefs.iniを
編集するためには、FireDAC Explorerを
[管理者として実行]が必要



BDEのエリアスの移行

FireDAC Explorerのメインメニュー [Connection]-[Import BDE Alias]を選択し、BDE Administratorに設定されている一部のエリアスをインポートできます。



- ・但し、Paradoxのエリアスはインポートできません
- ・エリアスの移行後、空欄のパラメータは個別の設定が必要

データベースドライバの対応表

データベース	BDE(SQL-Link)	FireDAC
Oracle	sqlora8.dll	TADPhysOracleDriverLink
MSSQLServer	sqlmss32.dll	TADPhysMSSQLDriverLink
DB2	sqldb2v5.dll	TADPhysDB2DriverLink
PostgreSQL	N/A	TADPhysPgDriverLink
MySQL	N/A	TADPhysMySQLDriverLink
SQLite	N/A	TADPhysSQLiteDriverLink
Sybase ASE	sqlsyb32.dll	TADPhysODBCDriverLink
Paradox	idpdx32.dll	N/A
ODBCブリッジドライバ	idodbc32.dll	TADPhysODBCDriverLink
DataSnap	N/A	TADPhysDataSnapDriverLink
InterBase FireBird	sqlint32.dll N/A	TADPhysIBDriverLink

FireDACでは、ターゲットのデータベースに応じた
TADPhysXXXXDriverLinkが必要

TSessionの代替は？

- アプリケーションでデータベース接続を管理する役割を持つコンポーネント

BDE	FireDAC
TSession	TADManager
Session変数	ADManager変数

- ADManagerへ接続設定を登録することで、ワークスレッド内で物理接続の再利用を行なうコネクションプーリングが利用できる

TTableを移行する最良な選択は？

● 普通にコンポーネントの名前だけで見ると、
TTable → TADTableへ置き換えたいところですが・・・

- ・TADTableは、利用に関して制限が多く、データの取得の際に余計なラウンドトリップが発生したり、オーバーヘッドが大きい
- ・TADQueryは、TTableと共通するプロパティやメソッドを持つ
- ・パフォーマンスは、TADQuery > TADTable

通常は、TADQueryへ置き換えることがベストな選択

データ型のマッピング

- ◎ FireDACは、dbExpressのデータ型マッピングに類似
- ◎ BDEからFireDACへ移行すると、RDBMSごとに異なるデータ型マッピングを持っているかもしれませんが、FireDACでは、データ型マッピングを調整するための強力な機能を備えています。

Format Options

▢ **Data Mapping Rules**

☒ Ignore inherited rules

SourceDataType	TargetDataType	PredMin	PredMax	ScaleMin	ScaleMax	SizeMin	SizeMax	NameMas
dtUInt16 ▼	dtInt32							

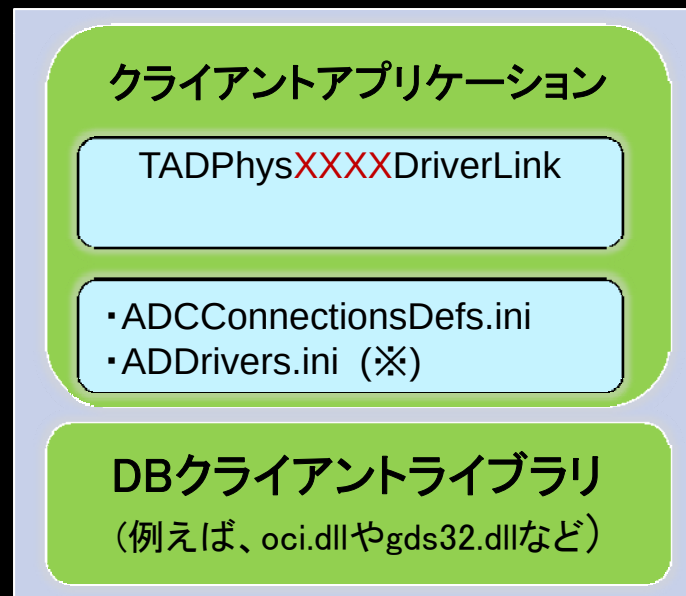
アプリケーションの配布

- ◎ FireDACは、BDEと比べて配布がとってもシンプル!!

BDEの場合:



FireDACの場合:



※もしFireDACのエリアスを参照していないアプリケーションの場合、この2つのファイルの配布は不要

互換性の無いBDEプロパティについて

- FireDACでは、一部 BDEに対応したプロパティを持っていない

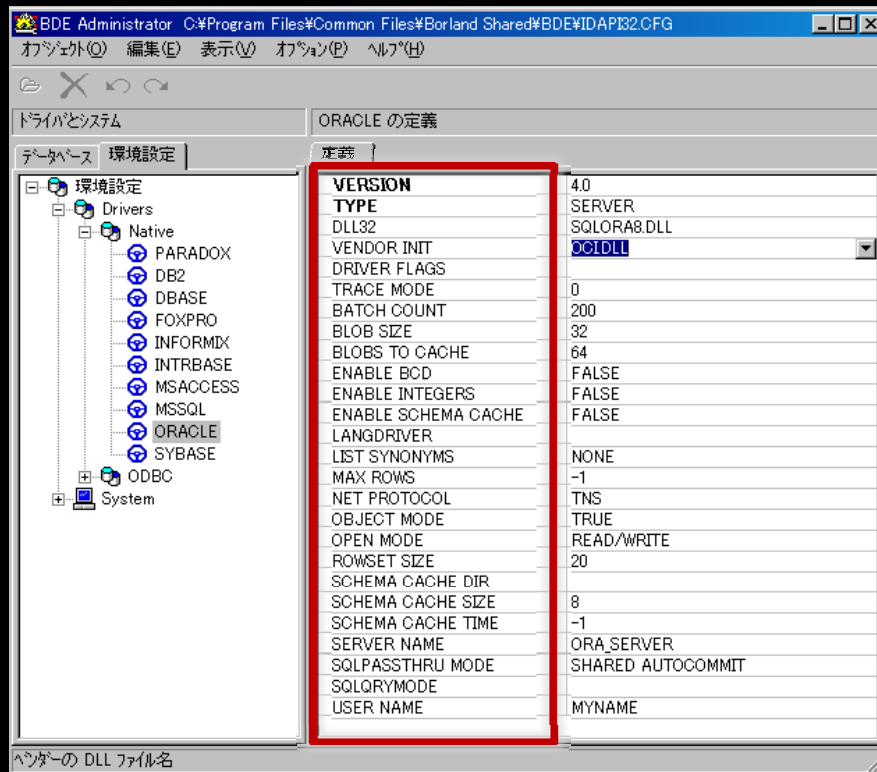
BDE プロパティ	FireDAC プロパティ
Alias	ConnectionDef
Database	Connection
SessionName	N/A
PrivateDir	N/A

※FireDACに存在しないプロパティは、完全に削除が必要

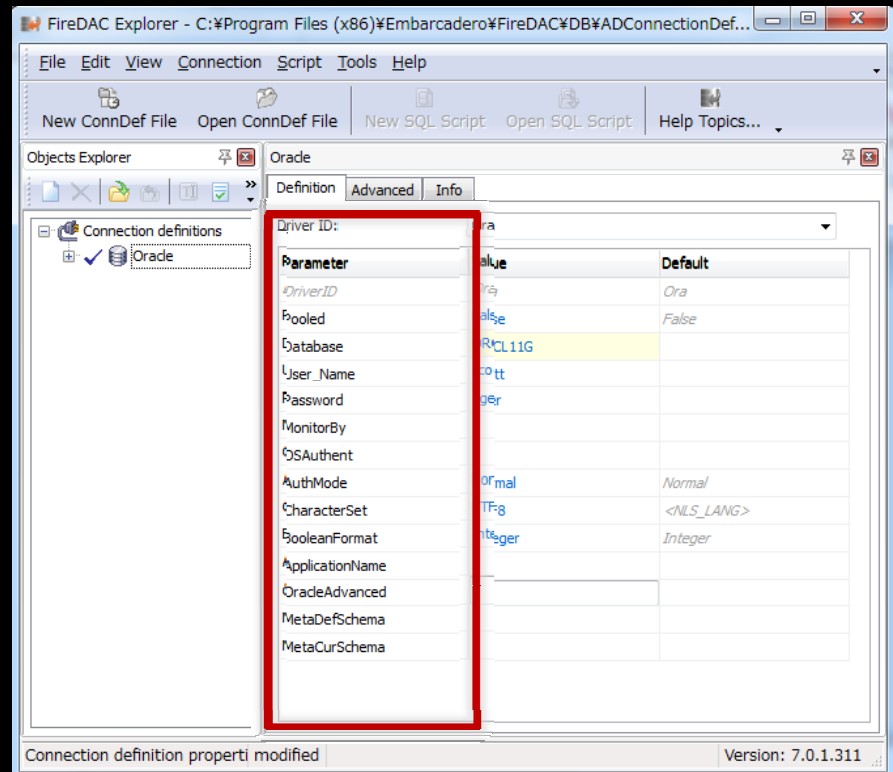
データベースドライバのパラメータの違い

下記は、Oracleドライバのパラメータ同士を比較した例:

BDE(SQLLink)



FireDAC



※パラメータ名が異なるため、個別に設定が必要になります

CharacterSet パラメータ

◎ FireDACでUnicodeデータベースに接続するには？

データベース	CharacterSet パラメータ
Oracle	utf8 あるいは NLS_LANG=_.UTF8
MSSQLServer	自動的にUTF-16がセットされる※
DB2	自動的にUTF-16がセットされる※
PostgreSQL	utf8
MySQL	utf8
SQLite	自動的にUTF-16がセットされる※
MS Access	自動的にUTF-16がセットされる※
InterBase /Firebird	utf8
Sybase SQL Anywhere	自動的にUTF-16がセットされる※
ODBCブリッジドライバ	自動的にUTF-16がセットされる※
dbExpress ブリッジドライバ	dbExpressドライバに依存

※ RAD Studio/Delphi/C++Builder のバージョンが2009以降の場合
2007以前のバージョンはUnicode未対応

OnLoginイベント

- TADConnection.OnLoginイベントは、TDatabase.OnLoginイベントのパラメータリストと互換性が無い

```
procedure TMyDataModule.dbLogin(Connection: TDEConnection; LoginParams:
TStrings);
begin
  LoginParams.Values['USER NAME'] := 'me';
  LoginParams.Values['PASSWORD'] := 'pwd';
end;
```

BDE



例として、以下のようにコードを置き換えれる

```
procedure TMyDataModule.dbLogin (AConnection: TADCustomConnection;
const AConnectionDef: IADStanConnectionDef);
begin
  AConnectionDef.UserName := 'me';
  AConnectionDef.Password := 'pwd';
end;
```

FireDAC

時間のかかる処理

```
Screen.Cursor := crSQLWait;  
try  
    .....  
finally  
    Screen.Cursor := crDefault;  
end;
```

BDE



例として、以下のようにコードを置き換えれる

```
uses  
    uADStanFactory, uADGUIxIntf;  
    .....  
var  
    oWait: IADGUIxWaitCursor;  
    .....  
ADCreateInterface(IADGUIxWaitCursor, oWait);  
oWait.StartWait;  
try  
    .....  
finally  
    oWait.StopWait;  
end;
```

FireDAC

データベースの例外エラーについて

BDEの例外クラス	FireDACの例外クラス
EDBEngineError	EADDBEngineException
エラーの種類を取得するには、 ErrorCode プロパティを使用	エラーを列挙した値を返す Kind プロパティを使用

```
if E is EDBEngineError then
begin
  case EDBEngineError(E).Errors[0].ErrorCode of
    DBIERR_KEYVIOL: MetaBaseDBError(SMb_DataSetInvalidPKeyValue, E);
  end;
```

BDE



以下のようにコードを書き換える

```
if E is EADDBEngineException then
begin
  case EADDBEngineException(E).Kind of
    ekUKViolated: MetaBaseDBError(SMb_DataSetInvalidPKeyValue, E);
  end;
```

FireDAC

注意点

- ◎ BDE API を直接呼び出しているプログラムコードは、そのままでは移行できない
 - ・BDE API(DBIxxxx)とFireDACのAPI(Phys interface)は、全く互換性が無いため、FireDACのAPIを利用して再コーディングが必要
- ◎ BDEとFireDACは異種クエリの構文と機能が異なる
 - ・FireDACでは、LocalSQLを利用し、BDEのような異種結合が行える機能を提供
- ◎ TBatchMoveとTADDataMoveは、機能的に多くの点で異なる（コードの書き直しが必要なケースあり）

BDEアプリを最新へ – FireDACによる移行

FireDACのデモ

デモ

- ◎ 簡単なBDEアプリからFireDACアプリへの移行
- ◎ パフォーマンス対決
dbExpressとFireDACの比較

BDEアプリを最新へ – FireDACによる移行

まとめ

FireDACへ移行することによって・・・

- ◎ 最新のOSやDBのバージョンに対応
- ◎ BDEと同様、共通手順で広範なDBへアクセス
- ◎ アプリケーションの配布がシンプルに
- ◎ BDEと同等以上のパフォーマンスが期待
- ◎ BDEとの互換性により、移行にかかる工数を最低限に抑えられる

さらにアプリケーション資産をニーズに合わせた
様々な分野への展開が期待できる



いつかは、BDEのアプリを
アップデートしなければなりません

では、それをいつやるのか？

今でしょ!

FireDACを利用するには？

- Delphi/C++Builder/RAD Studio XE3
 - Enterprise Editionライセンスに含まれる
 - 既にXE3を購入済みのユーザーも含む
- FireDAC Client/Server Pack for XE3
 - Professional Editionに追加するアドオン

FireDACを無料で入手！	FireDAC Client/Server Pack
Delphi、C++Builder、RAD Studio XE3 Enterprise、Ultimate、Architect版のいずれかを購入 まず、購入したXE3製品をインストールして製品登録を行ってください。次に、こちらをクリックしてFireDACを無料ダウンロードします。	Delphi、C++Builder、RAD Studio XE3 Professional版を購入された方は、FireDAC Client/Server Add-On Packを特別価格で購入いただけます。 購入・お問い合わせはこちら

参考情報

- FireDACのオンラインヘルプ(英語)
http://docs.embarcadero.com/products/rad_studio/firedac/
- RAD Studioマイグレーションセンター
<http://www.embarcadero.com/jp/rad-in-action/migration-upgrade-center>

Thank you!