【A1】Delphi/C++テクニカルセッション

旧Delphiアプリケーション移行の実際例

～ 実例から学ぶ改修のポイント ～

人と技術をつなぐ

2016年12月9日
田中 芳起

自己紹介

名前：田中 芳起（たなか よしき）

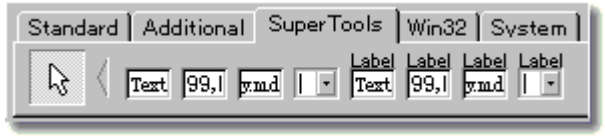
- 中堅SIerでパッケージシステムの開発／プロジェクト管理／品質管理 等の仕事に従事
- Delphiとは 1.0US版 からの付き合い

- デブキャンプでの講演実績
  - 26th：はじめてのFireDAC
  - 29th：Delphiで作るデータベースツール
  - 30th：今さら聞けない!? FireDAC入門
  - 33th：旧Delphiアプリケーション移行の実際例

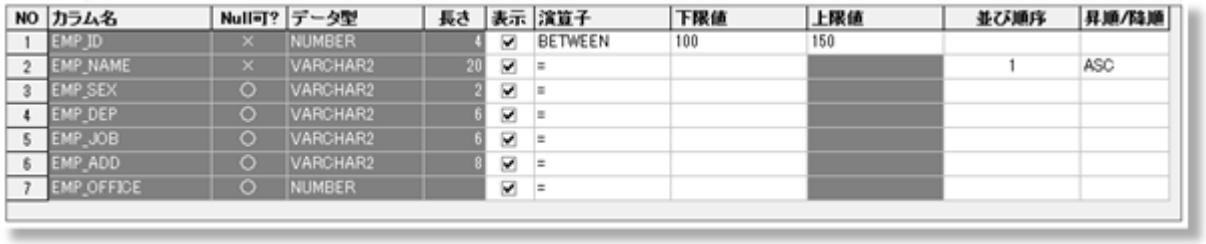
- ホームページ：http://www.avsoft.jp/
- Facebook：https://www.facebook.com/yoshiki.tanaka.942/
  - ：https://www.facebook.com/VisualNavi
  - ：https://www.facebook.com/groups/864814060228437/

コンポーネントの開発・販売

- **SuperEdit（入力系コンポーネント）**



- **SuperGrid（グリッド・コンポーネント）**



NO	カラム名	Null可?	データ型	長さ	表示	演算子	下限値	上限値	並び順序	昇順/降順
1	EMP_ID	×	NUMBER	4	☑	BETWEEN	100	150		
2	EMP_NAME	×	VARCHAR2	20	☑	=			1	ASC
3	EMP_SEX	○	VARCHAR2	2	☑	=				
4	EMP_DEP	○	VARCHAR2	6	☑	=				
5	EMP_JOB	○	VARCHAR2	6	☑	=				
6	EMP_ADD	○	VARCHAR2	8	☑	=				
7	EMP_OFFICE	○	NUMBER		☑	=				

・大手コンビニや、多くの企業システム構築で使用されてきたVCLコンポーネント
・6月から販売を開始

▶ ご購入はこちらから

O2 Components for Delphi

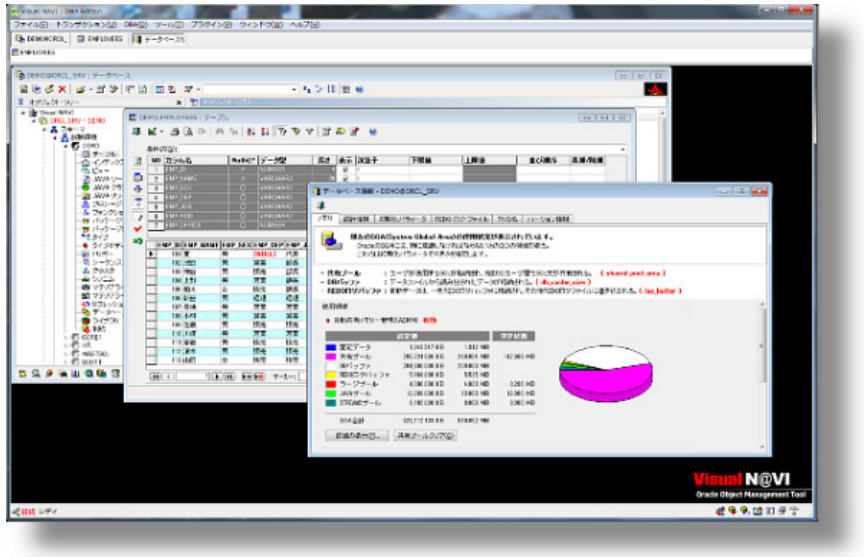
株式会社オン・アンド・オン
http://o2components.on-and-on.biz/

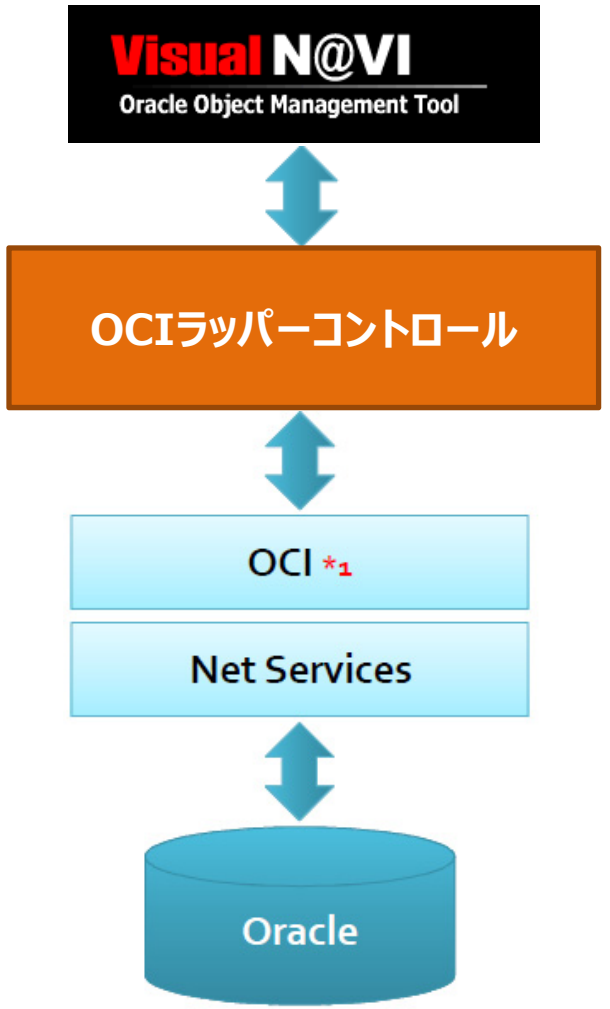
ComponentSource

株式会社コンポーネントソース
https://www.componentsource.co.jp/product/superedit-j
https://www.componentsource.co.jp/product/supergrid-j

統合型開発支援ツールの開発・販売

- **Visual NAVI**

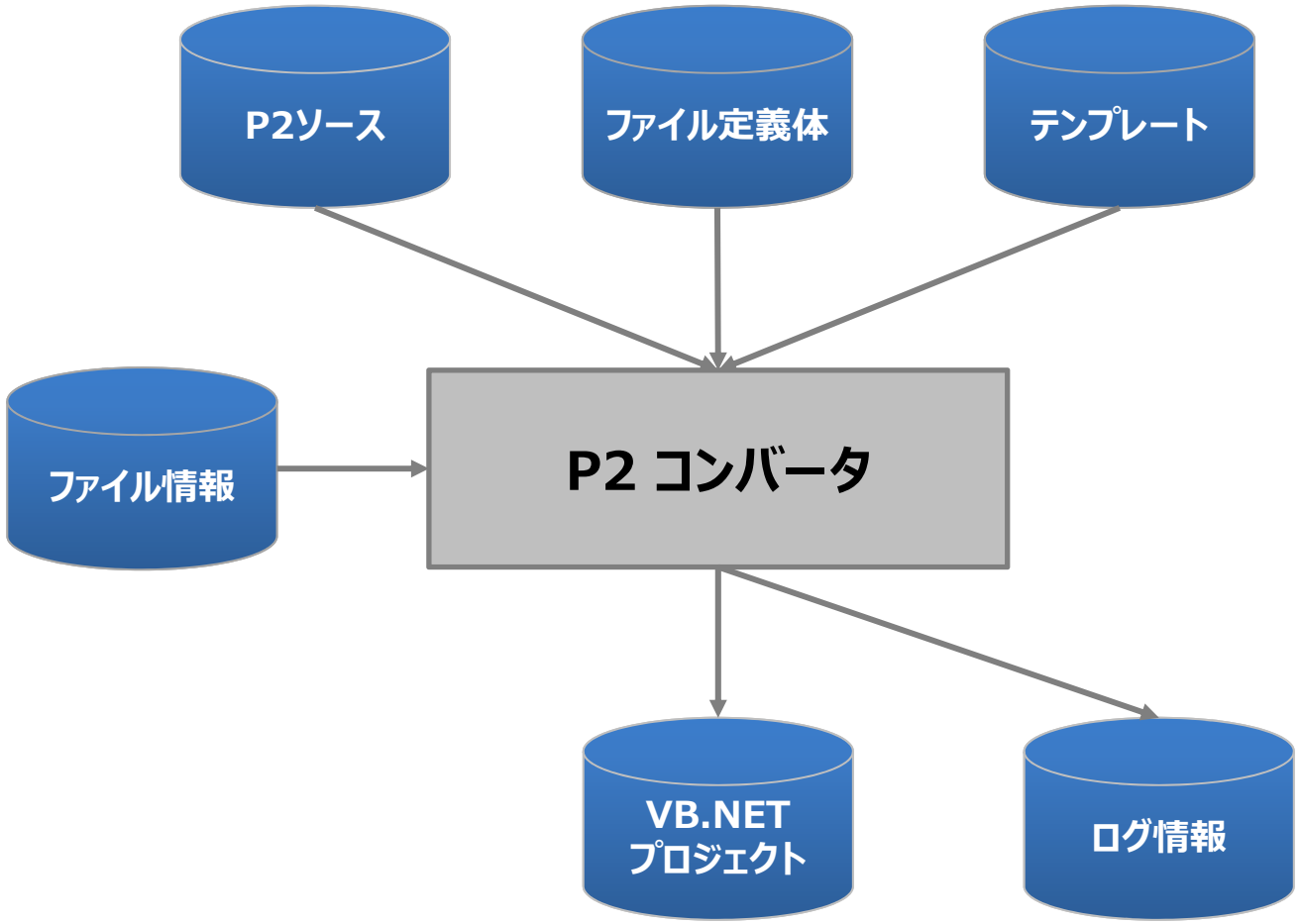




・Oracleアプリケーション開発支援
・GUIによるデータベース管理機能
・統合型開発支援ツール
・オラクル社のOCIを使用した高速接続を実現

***1 OCI**（**O**racle **C**all **I**nterface）は、Oracle社が提供するAPIです
※ Facebookページで情報発信しています。(https://www.facebook.com/VisualNavi)

移行実績（プログレスII *1 → Visual Basic.NET）



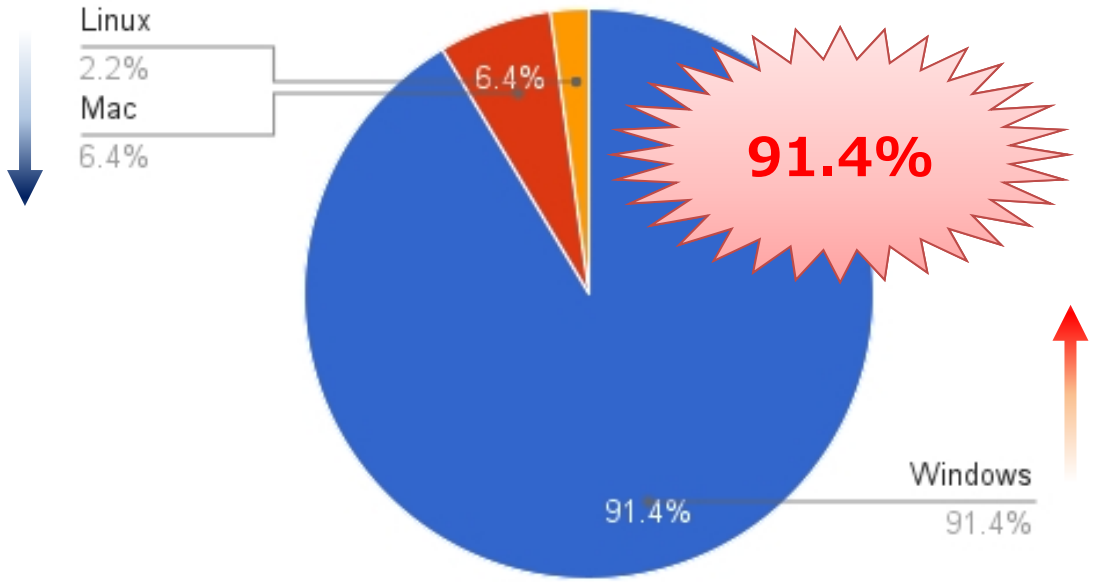
- プログラム本数： 1,597 本
- 総ステップ数：512,867 Step

*1 三菱電機が開発したオフィス業務におけるさまざまな処理を簡単に書き表せる簡易プログラム言語。IBMのRPGに類似している。

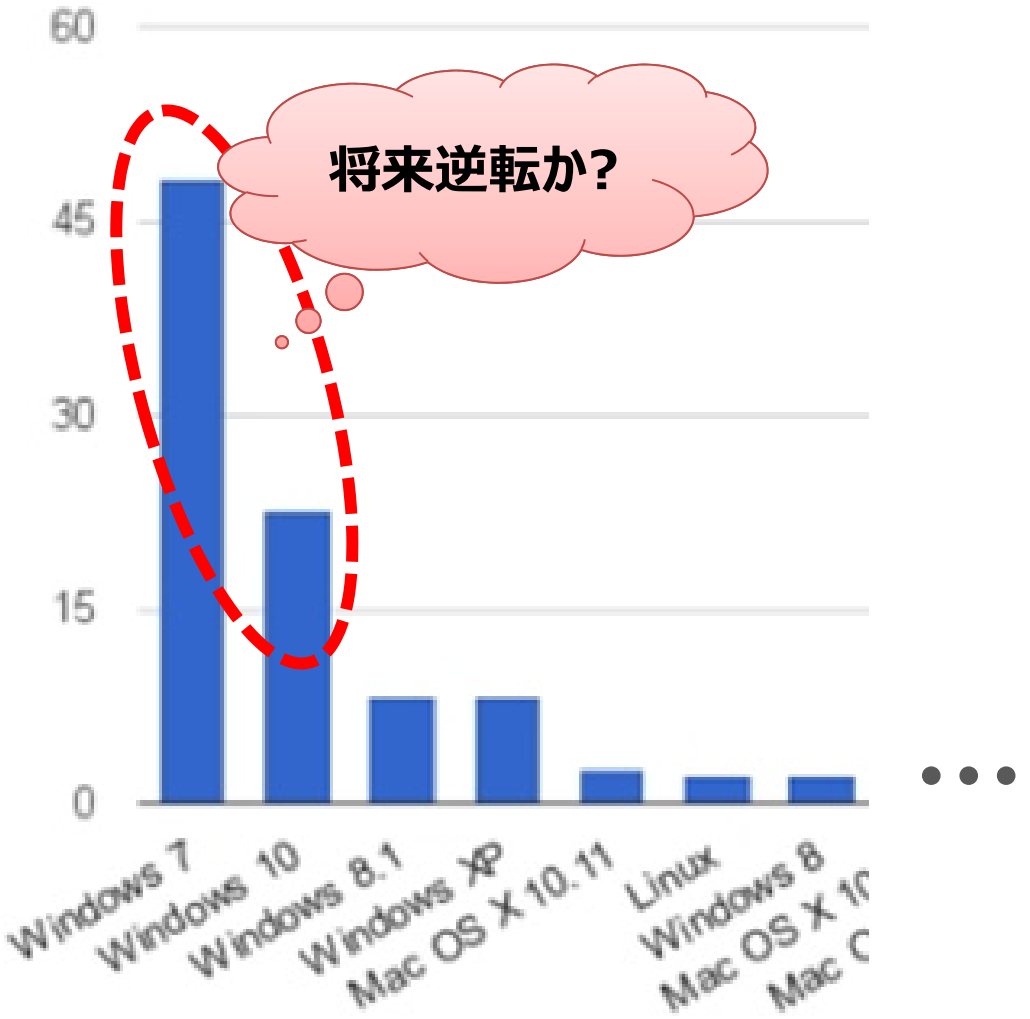
アジェンダ

- **はじめに**
- 移行の詳細
- ツールによる移行
- 移行後の対応
- XenApp サーバーを使う場合の注意点
- まとめ

デスクトップOSシェアは？

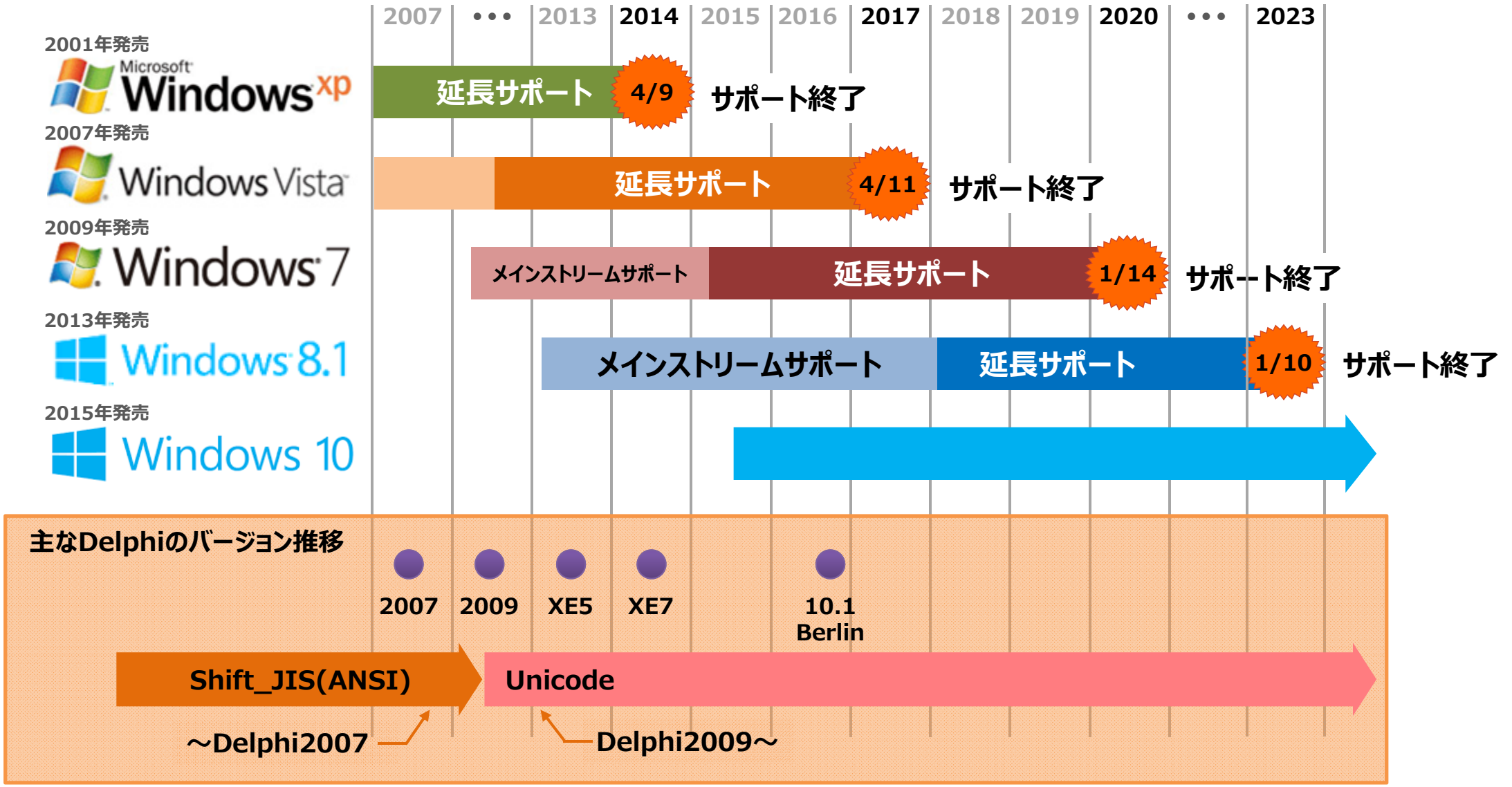


2016年10月



出典：Net Applications (https://netmarketshare.com/)

Windowsサポートライフサイクルと、Delphiバージョンの推移



移行のポイント１ ・・・ どのバージョンに移行するか?

- 古いプロジェクトを最新Delphi(Unicode版)でコンパイルすると様々な問題に直面する
- 段階的にプロジェクトをアップグレードすると、問題の切り分けが容易になる
- 一足飛びでプロジェクトをアップグレードすると様々な要因が絡み合い、バグなのか仕様なのかを特定することさえ困難になってしまう

- Vistaに対応しており、Altキー問題も発生しない
- Unicode版Delphiに限りなく近いため、Delphi2007からのマイグレーションだと基本的に文字コード絡みの問題しか起きない

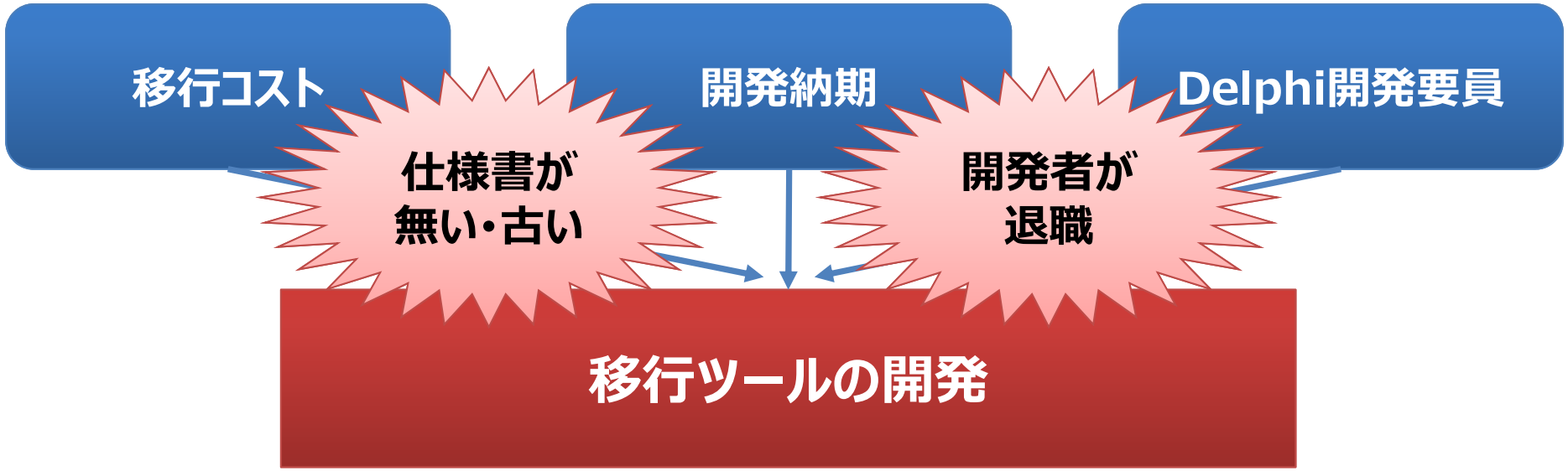
移行のポイント２ ・・・ コンポーネントは？

- **データベースクライアントAPIの変更**
  - **Borland Database Engine (略称: BDE)**
    長年Delphiに標準搭載されたデータベースクライアントAPI
    Delphi 6 付属の5.2が最新
  - **dbExpress (略称: DBX)**
    FireDAC登場以前では主力であったデータベースクライアントAPI
  - **dbGo (旧称 ADO Express)**
    WindowsのOLE DBインタフェースを利用したデータベースクライアントAPI
  - **FireDAC**
    XE4でバンドルされ(AnyDAC)、XE5で標準搭載されたデータベースクライアントAPI

- **サードパーティ製コンポーネントの移行**
  - **入力／グリッド**
    SuperEDIT／SuperGRID／TMS Grid ・・・
  - **帳票ツール**
    Quick Report／RaveReport／FastReport ・・・

※RAD Studioマイグレーションセンター
https://www.embarcadero.com/jp/rad-in-action/migration-upgrade-center

移行の課題と移行ツールのメリットは？



移行ツールを使うメリット

- レガシーシステムの機能をそのまま、新システムに移行できる
- 移行時のコストを大幅に抑え、旧システムの資産をほぼそのまま活用できる
- 新システム開発導入にともなう開発リスクも大幅に低減
- 現行システムの品質をほぼ継承できる（高品位なシステム移行を実現）
- システム変更にともなうSEの教育費用・期間が掛からない

移行システムの概要

- 環境比較（サーバー／データベース／開発ツール）

旧システム

- Windows Server 2003 (32bit)
- Oracle Server 10.2.0.1 (32bit)
- Delphi 5.0 (32bit)
- QuickRepor 3.0.7
- BDE 5.2

新システム

- Windows Server 2012 R2 (64bit)
- Oracle Server 11.2.0.4 (32bit)
- Delphi 2007 (32bit)
- QuickRepor 4.6
- FireDAC 7.0.1.3119

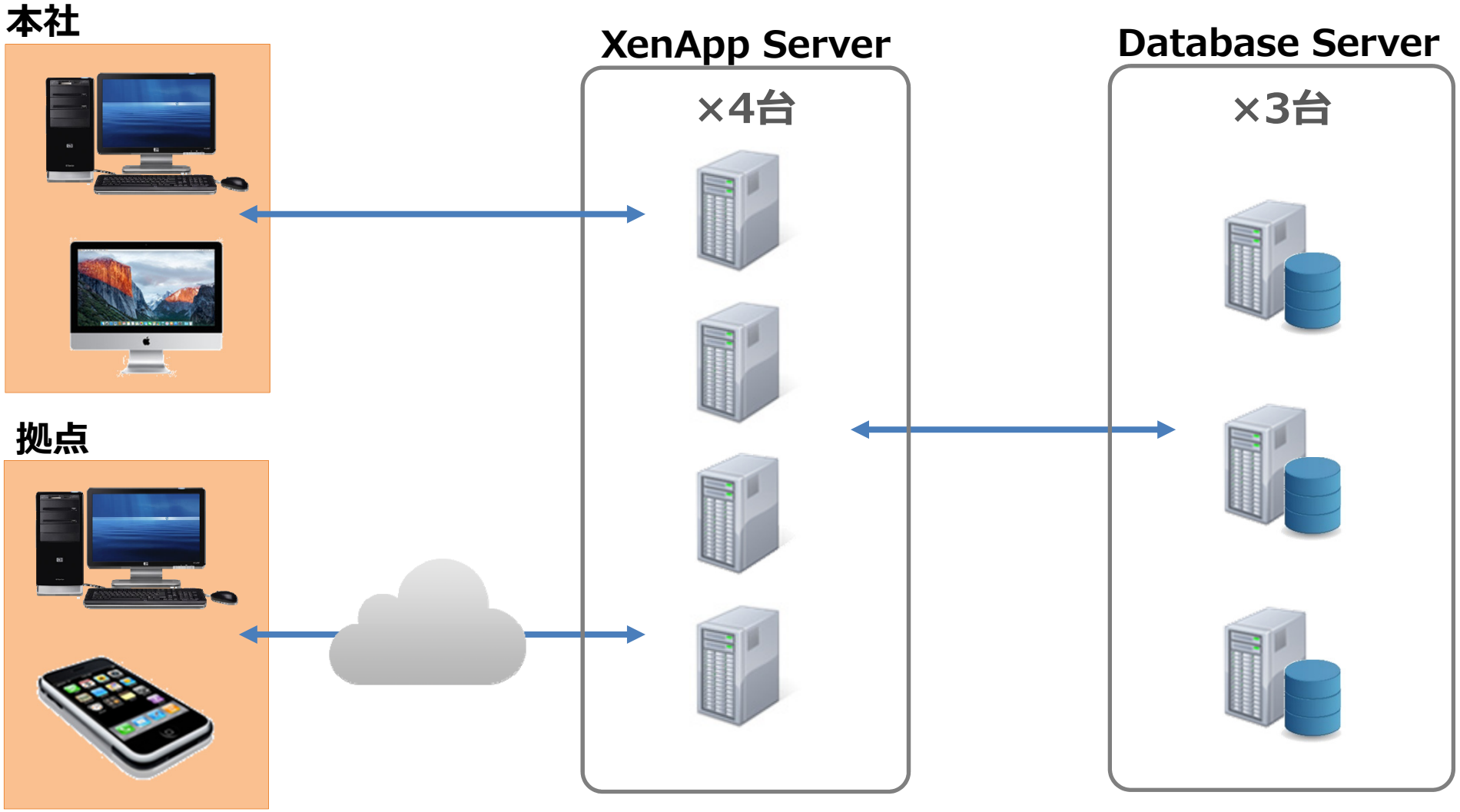
- システム規模

分類	機能数
入力	17
帳票	114
保守	63
バッチ	15
その他	73
合　計	282

- プログラム本数： 445 本
- 総ステップ数： 481,496 Step

新システム構成図

Citrix XenApp サーバーを使ったアプリケーションの仮想化

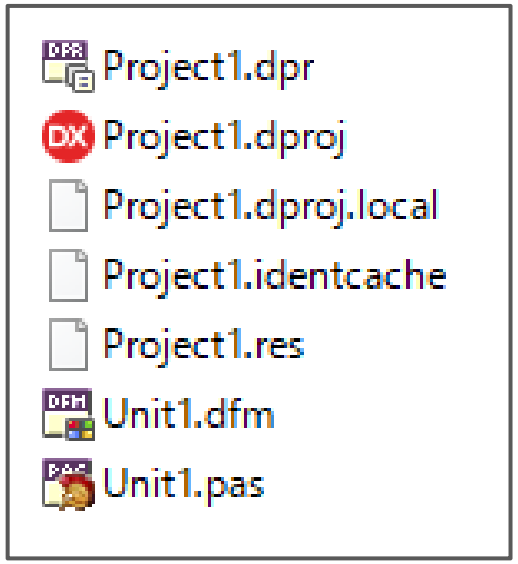


アジェンダ

- はじめに
- **移行の詳細**
- ツールによる移行
- 移行後の対応
- XenApp サーバーを使う場合の注意点
- まとめ

Delphiプロジェクトファイルの構成

- プロジェクトファイル群



- Project1.dpr

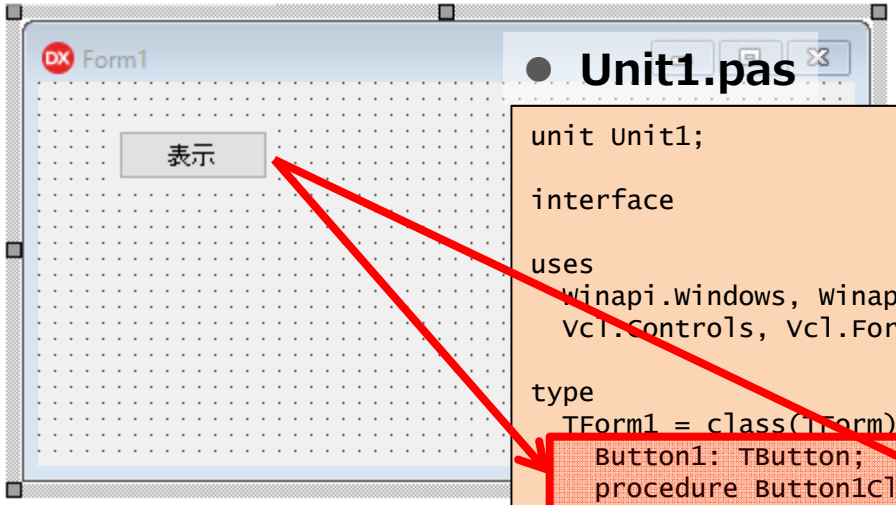
```
program Project1;

uses
  Forms,
  Unit1 in 'Unit1.pas' {Form1};

{$R *.res}

begin
  Application.Initialize;
  Application.MainFormOnTaskbar := True;
  Application.Title := 'テストプログラム';
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.
```

- 拡張子の説明

拡張子	説　　明
.dpr	Delphi用プロジェクトファイル（旧）
.dproj	Delphi用プロジェクトファイル。中身は XML（新）
.local	ユーザー固有のプロジェクトオプションを格納
.identcache	リファクタリング用の（キャッシュ）ファイル
.res	バージョン情報やアイコン等が埋め込まれたリソースファイル
.dfm	VCL フォームのファイル。テキストとバイナリ形式がある。（*.pasと対で作成される）
.pas	ソースコードファイル

ファイルの中身は・・

- TForm1



- Unit1.pas

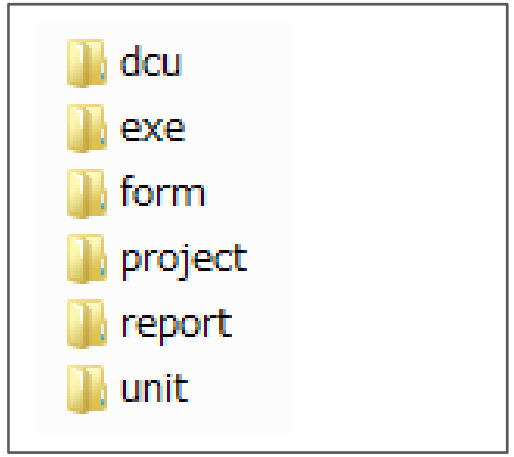
```
unit Unit1;

interface

uses
  Winapi.Windows, Winapi.Messages, System.SysUt
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Std

type
  TForm1 = class(TForm)
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  private
    { Private 宣言 }
  public
    { Public 宣言 }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin
  ShowMessage('Hello world');
end;

end.
```

- Unit1.dfm

```
object Form1: TForm1
  Left = 0
  Top = 0
  Caption = 'Form1'
  ClientHeight = 192
  ClientWidth = 408
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = []
  OldCreateOrder = False
  PixelsPerInch = 96
  TextHeight = 13
  object Button1: TButton
    Left = 40
    Top = 24
    Width = 75
    Height = 25
    Caption = #34920#31034
    TabOrder = 0
    OnClick = Button1Click
  end
end
```

移行システムのフォルダ構成

- このシステムは、次の６個のフォルダで管理される



- フォルダの説明

フォルダ名称	内　　容
dcu	コンパイル済ユニットファイル
exe	コンパイルされた実行モジュール
form	画面のフォームファイル(*dfm)／ソースファイル(*.pas)
project	プロジェクトファイル
report	帳票のフォームファイル(*dfm)／ソースファイル(*.pas)
unit	共通ユニット

移行の手順と概要

Step1	フォームファイル の変換
Step2	プロジェクトファイルと関連ファイルのコピー
Step3	データベース･コンポーネントの分離
Step4	プロジェクトファイルの修正
Step5	フォームファイル の修正
Step6	ソースファイル の修正

フォームファイル (*.dfm) の変換

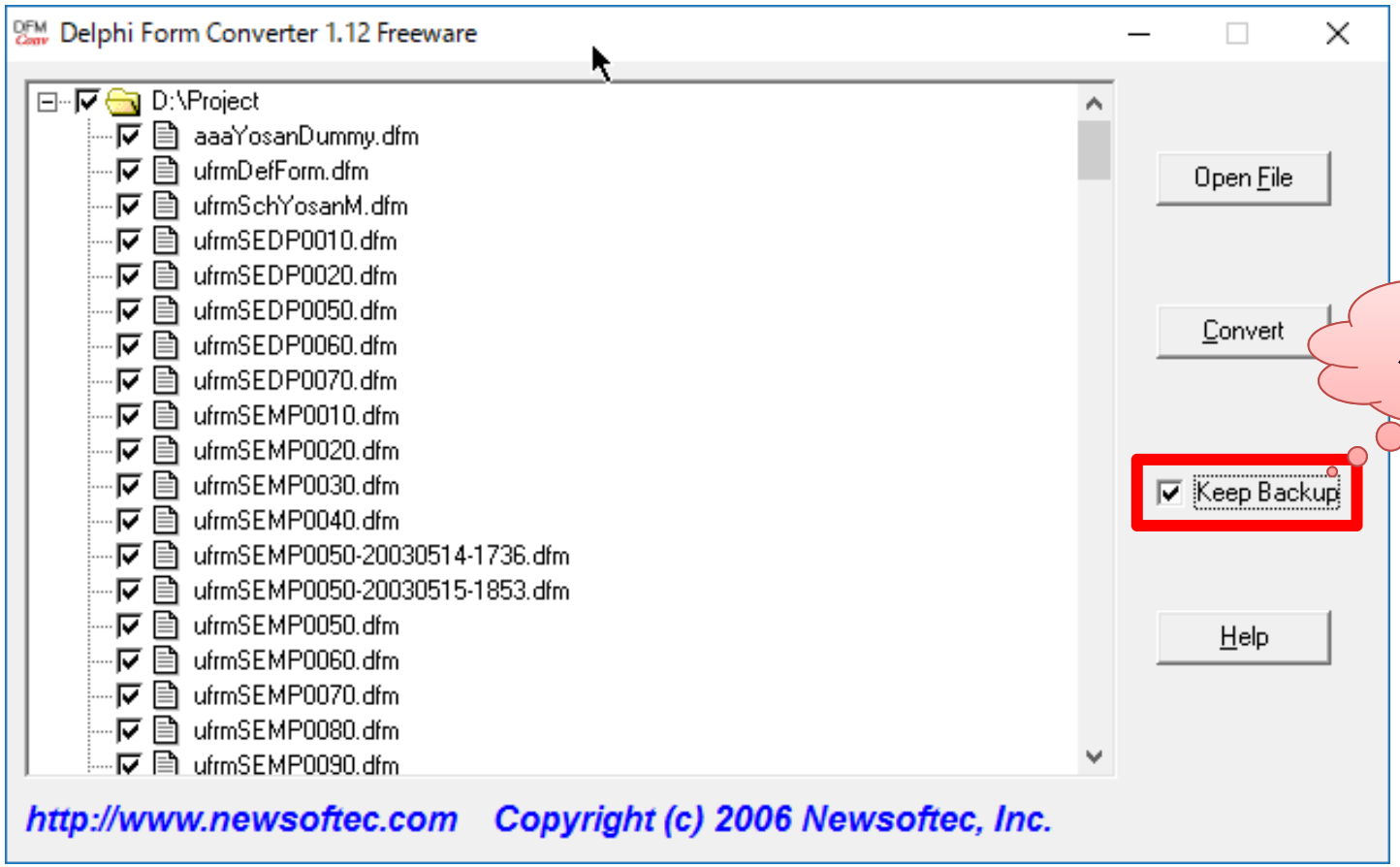
- バイナリエディタで先頭バイトを判定することにより可能

```
ADDRESS   00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F   0123456789ABCDEF
00000000  FF 0A 00 54 52 50 54 53 45 4D 50 30 30 35 31 00   ...TRPTSEMP0051.
00000010  30 10 30 6C 00 00 54 50 46 30 0C 54 72 70 74 53   0.0l..TPF0.TrptS
00000020  45 4D 50 30 30 35 31 0B 72 70 74 53 45 4D 50 30   EMP0051.rptSEMP0
00000030  30 35 31 04 4C 65 66 74 02 00 03 54 6F 70 02 00   051.Left...Top..
00000040  05 57 69 64 74 68 03 3A 05 06 48 65 69 67 68 74   .Width.:..Height
```

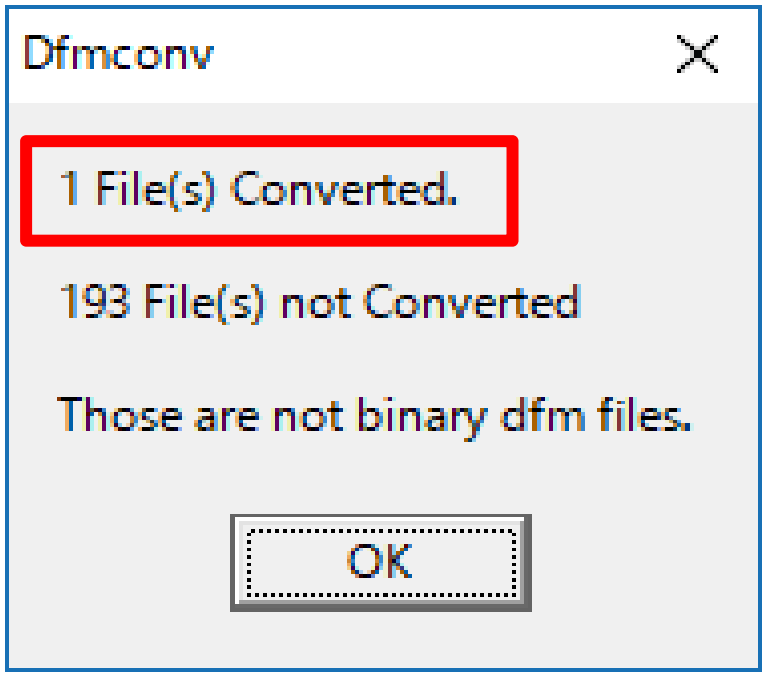
先頭3バイトが「FF 0A 00」は、バイナリ形式のフォームファイル

- Delphiのインストールフォルダ¥bin にある「convert.exe」を利用し、フォームファイルをバイナリからテキスト形式に相互変換することができる

```
convert.exe -i -s -t *.dfm
```



バックアップ



※ツールの入手先 http://www.brothersoft.com/delphi-form-converter-download-48912-s1.html

プロジェクトファイルと関連ファイルのコピー

- 旧システムの「project」フォルダから、該当プログラムの次の拡張子のファイルを新しい開発環境にコピーする

*.dpr
*.res

- プロジェクトファイル(*.dpr)をテキストエディタで開く

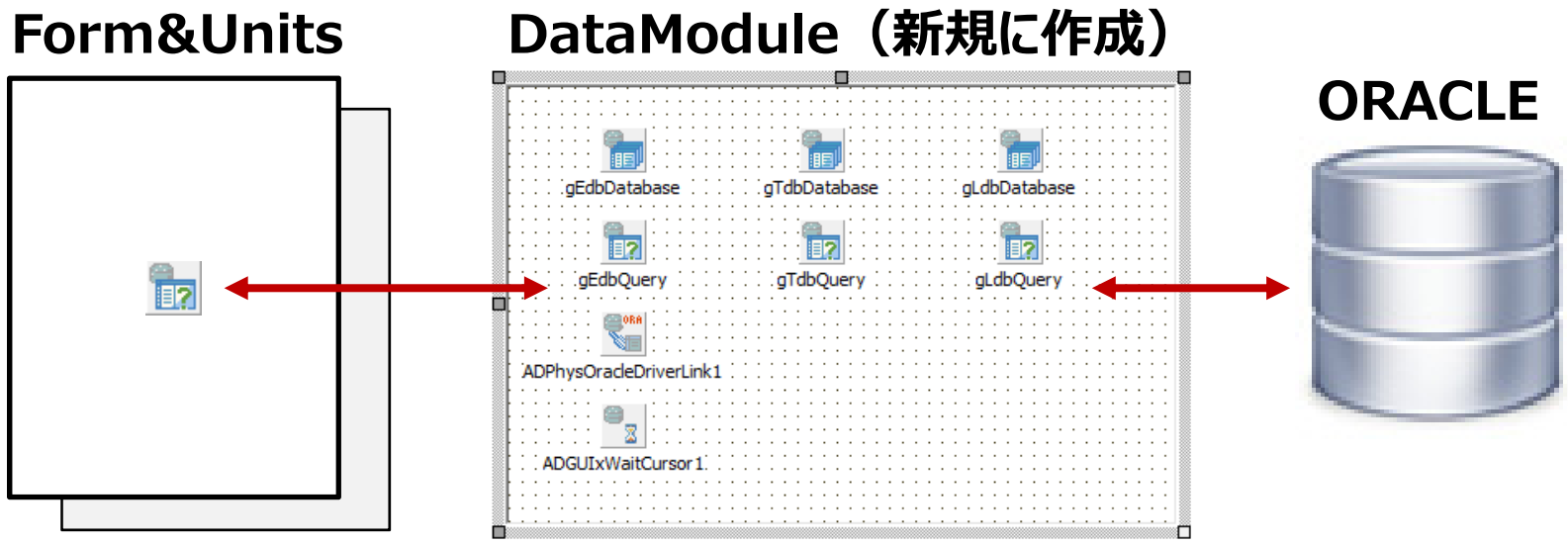
```
uses
  Forms,
  uInstance,
  ufrmSplash,
  ufrmSEMP0010 in '..¥form¥ufrmSEMP0010.pas' {frmSEMP0010},
  urptSEMP0010 in '..¥report¥urptSEMP0010.pas' {rptSEMP0010: TQuickRep};

{$R *.RES}
```

相対パス指定

ユニット参照

- プロジェクトに組み込まれている関連ファイル(*.pas／*.dfm)を新しい開発環境のフォルダに全てコピーする

データベース･コンポーネントの分離



データモジュールを使うメリット

- データベース接続とビジネスルールを完全に分離できる
- データベース処理を纏めることができる
- プラットフォーム非依存が可能
- 将来多層化することが可能

プロジェクトファイル (*.dpr) の修正

- **uses節のユニット参照(in) 先頭に次の１行を追加する**

DataModule in '..¥unit¥DataModule.pas' {dmOracle: TDataModule},

```
uses
  Forms,
    :
  DataModule in '..¥unit¥DataModule.pas' {dmOracle: TDataModule};
{$R *.RES}

begin
  try
    if gfCreateInstance('TfrmSEDP0010') then
    begin
      Application.Initialize;
      Application.Title := 'ＸＸＸＸ原票入力';
      Application.CreateForm(TdmOracle, dmOracle);
      Application.CreateForm(TfrmSEDP0010, frmSEDP0010);
      Application.Run;
    end;
    gpReleaseInstance;
  except
    {----例外処理を記述 ----}
  end;
end.
```

フォームファイル (*.dfm) の修正

- **TDatabase の削除**

```
object Database1: TDatabase
    Connected = True
    DatabaseName = 'db1'
    DriverName = 'ORACLE'
    LoginPrompt = False
    Params.Strings = (
      'SERVER NAME=····'
      'USER NAME=···· '
      'PASSWORD=····')
    SessionName = '···· '
    Left = 128
    Top = 28
  end
```

- **クラス名の変更**

TQuery	→ TADQuery
TLblText	→ TLabeledText
TLblNumber	→ TLabeledNumber
TLblDate	→ TLabeledDate
TLblComboBox	→ TLabeledComboBox

- **上記コンポーネントで不要となったプロパティを削除**

- **DatabaseName／SessionNameプロパティの削除（TADQueryでは、不要となった）**

```
object QueryReport: TQuery
    DatabaseName = 'STAXDEBUG'
    SessionName = 'Default'
    SQL.Strings = (
      'select *'
      'from WKSYGL0020 a'
      'order by a.ブロックNO, a.出力順'
      '')
    Left = 56
    Top = 12
  end
```

- **Zoomプロパティの次に、以下の3行を追加する（QuickReport の不具合対応）**

```
  Units = MM
  Zoom = 100
  PreviewInitialState = wsMaximized
  PrevShowThumbs = False
  PrevShowSearch = False
  object ColumnHeaderBand1:  TQRBand
```

ソースファイル (*.pas) の修正

- uses節(interface部)の最後に、uADCompClient ユニットを追加する *1

```
interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  DBTables, StdCtrls, Buttons, ASCtrls, Label3D, IniFiles, uADCompClient;

type
```

*1 但し、ソースファイルにTQuery（TADQuery）が定義されているときのみ追加する。

- uses節(implementation部)の最後に、DataModule ユニット名を追加する

```
implementation
   :
uses
  uOracleObj, uPRGstatus, ufrmPrintDialog, ufrmSchM, ufrmSchHojo,
  ufrmSelectDialog, DataModule;
```

- **TDatabase の削除**
- **クラス名の変更**

TQuery → TADQuery
TLblText → TLabeledText
TLblNumber → TLabeledNumber
TLblDate → TLabeledDate
TLblComboBox → TLabeledComboBox

TADQueryのインスタンスを生成

TADQueryのプロパティ設定／TADConnectionの関連付け

- **TQuery.Createを「CreateQuery」に書き換える**
- **「SetQueryProperty」を挿入する**

```
var
  FQuery: TQuery;
begin
  FQuery := TQuery.Create(nil);
```

```
var
  FQuery: TADQuery;
begin
  FQuery := CreateQuery;
  SetQueryProperty(FQuery, gEdbDatabase);
```

※ CreateQuery／SetQueryPropertyは、データモジュール(DataModule.pas)に定義されている。

- **SetQueryProperty の引数（TADQueryとTADConnection）を記入する**
- **QueryReportの SessionName／DatabaseName をコメント行（削除）にする**

```
try
  with rptSEDP0050.QueryReport do
  begin
    Active := False;
    SetQueryProperty(rptSEDP0050.QueryReport, gEdbDatabase);
    SQL.Clear;
    SQL.Add('select * from ' + lstrPrtWk + ' a ');
    SQL.Add('order by ');
    SQL.ADD('a.営業所コード,a.部課コード,a.通番,a.行');
//  SessionName := gEdbDatabase.SessionName;
//  DatabaseName := gEdbDatabase.DatabaseName;
    Active := True;
  end;
  Ffrm.ShowModal;
finally
  Ffrm.Release;
end;
```

大変な作業

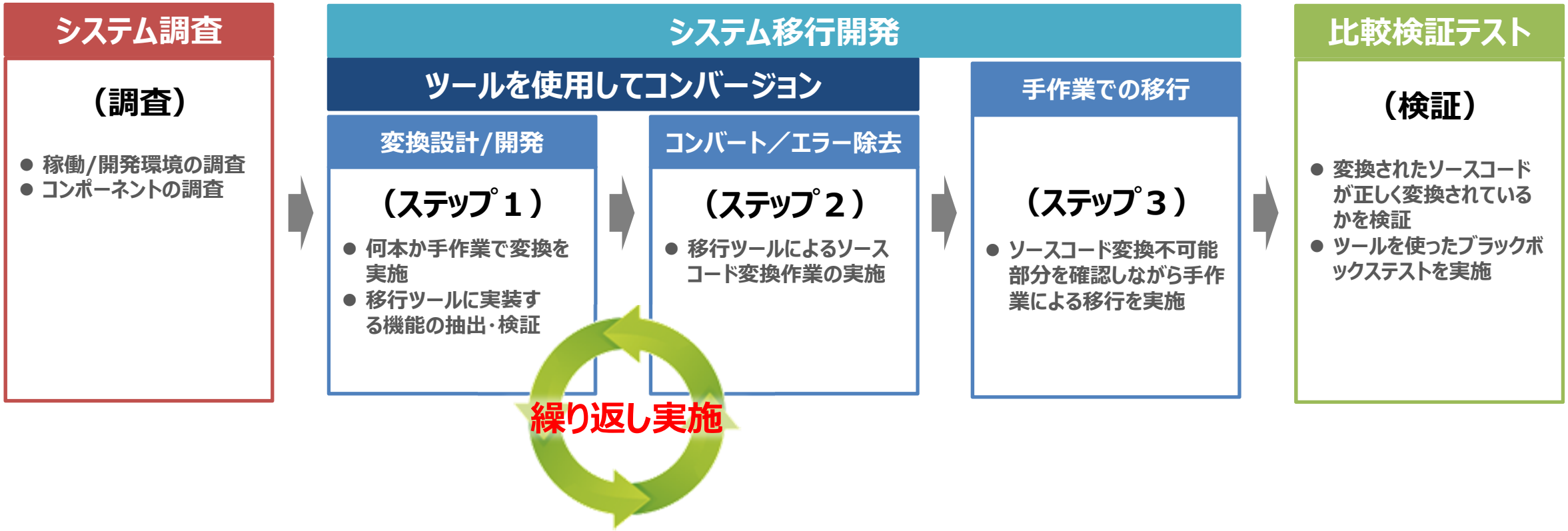
- **バインド変数名の変更（SQL実行時エラーの場合）**

BDEではバインド変数に漢字を使用してもSQLエラーとならなかったが、FireDAC 7.xは完全なUnicode対応が行われていないため、バインド変数名に漢字が使われていると実行時エラーとなる場合がある

アジェンダ

- はじめに
- 移行の詳細
- **ツールによる移行**
- 移行後の対応
- XenApp サーバーを使う場合の注意点
- まとめ

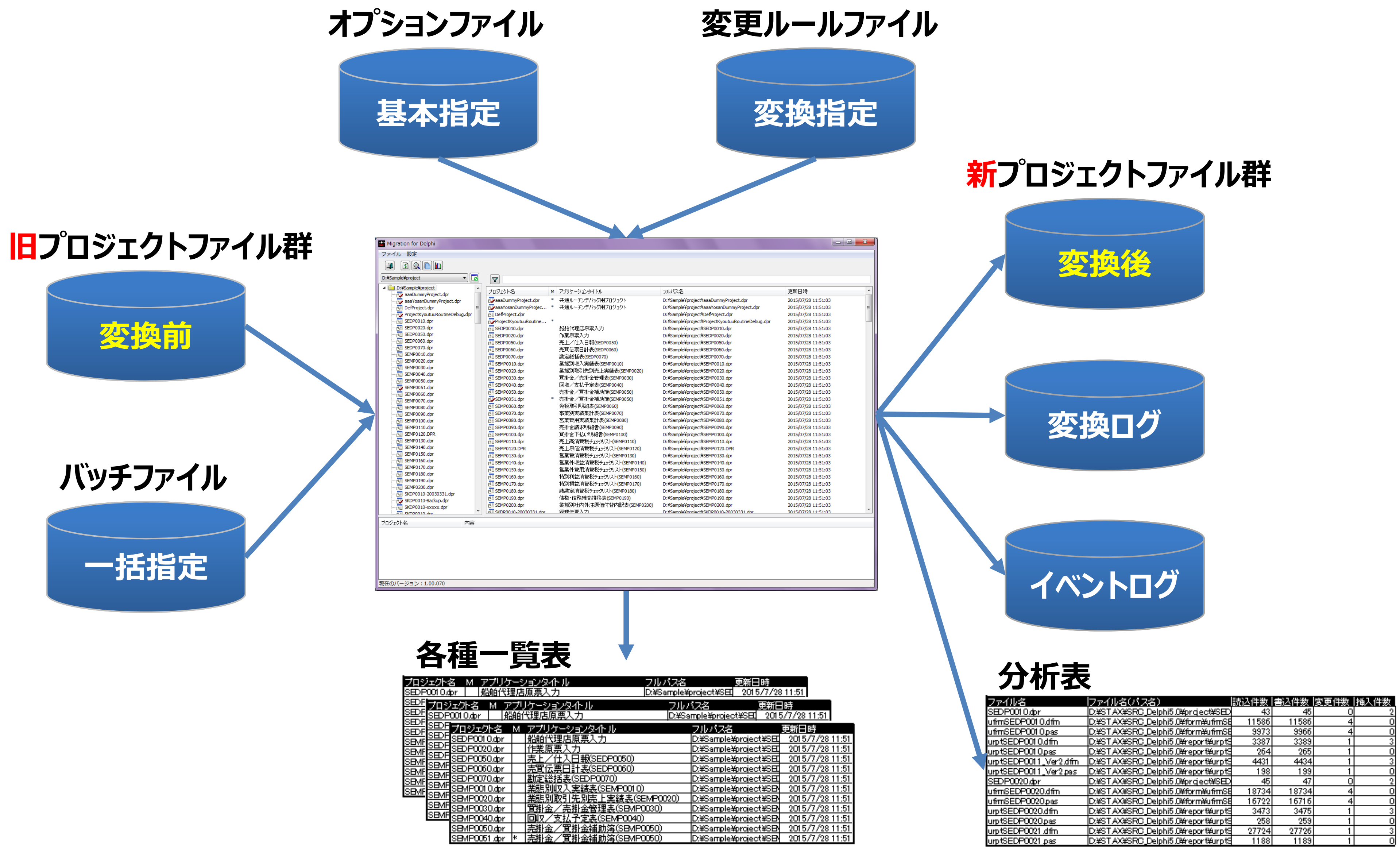
移行作業の流れ



システム移行のポイント

- ツールですべて変換しようと思わない！
- 手作業が必要な箇所は、あえてエラーになる様な変換をする
- 比較検証は、ツールを使った「ブラックボックステスト」を行う

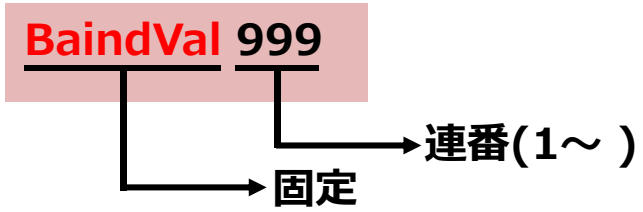
移行ツールの概要



バインド変数名の変換処理

BDEではバインド変数に漢字を使用してもSQLエラーとならなかったが、FireDAC 7.x は完全なUnicode対応が行われていないため、バインド変数名に漢字が使われていると実行時エラーとなる場合がある

- **コンバータでは、定義されているバインド変数名を次の通り変換する**




```
SQL.Add('SELECT ');
  :
SQL.Add('摘要コード = :P_摘要コード');

// パラメータ指定
ParamByName('P_摘要コード').AsInteger := AbstractCode1;
```

```
SQL.Add('SELECT ');
  :
SQL.Add('摘要コード = :BindVal1');

// パラメータ指定
// BaindVal1: P_適用コード
ParamByName('BindVal1').AsInteger := AbstractCode1;
```

変換コードサンプル

変換前のDelphi5のコード

```
SQL.Add('INSERT INTO ' + lStrTblName);
SQL.Add('(');
SQL.Add('PAGENO, ');
SQL.Add('摘要1, ');
SQL.Add('請求明細NO, ');
SQL.Add('金額合計, ');
SQL.Add('消費税合計, ');
SQL.Add('総合計');
SQL.Add(') ');
SQL.Add('VALUES ');
SQL.Add('(');
SQL.Add(':P_PAGENO, ');
SQL.Add('NULL, ');
SQL.Add('99, ');
SQL.Add(':P_金額合計, ');
SQL.Add(':P_消費税合計, ');
SQL.Add(':P_総合計');
SQL.Add(') ');

ParamByName('P_PAGENO').AsInteger := 1;
ParamByName('P_金額合計').AsCurrency := curKin1;
ParamByName('P_消費税合計').AsCurrency := curKin2;
ParamByName('P_総合計').AsCurrency := curKin3;

for linti := intRecCnt + 1 to 20 do
  ExecSQL; //SQL文を実行

SQL.Clear;

if Params.Count <> 0 then Params.Clear; //Paramコレクション

SQL.Add('UPDATE ' + lStrTblName + ' SET ');
SQL.Add('(');
SQL.Add('請求年, ');
SQL.Add('請求月, ');
SQL.Add('請求日, ');
```

行: 7346 列: 1/1 文字: 1/1 コードページ: 932(shift_jis) RO Win

変換後のDelphi2007のコード

```
SQL.Add('INSERT INTO ' + lStrTblName);
SQL.Add('(');
SQL.Add('PAGENO, ');
SQL.Add('摘要1, ');
SQL.Add('請求明細NO, ');
SQL.Add('金額合計, ');
SQL.Add('消費税合計, ');
SQL.Add('総合計');
SQL.Add(') ');
SQL.Add('VALUES ');
SQL.Add('(');
SQL.Add(':BindVal79, ');
SQL.Add('NULL, ');
SQL.Add('99, ');
SQL.Add(':BindVal80, ');
SQL.Add(':BindVal81, ');
SQL.Add(':BindVal82');
SQL.Add(') ');

ParamByName('BindVal79').AsInteger := 1;
ParamByName('BindVal80').AsCurrency := curKin1;
ParamByName('BindVal81').AsCurrency := curKin2;
ParamByName('BindVal82').AsCurrency := curKin3;

for linti := intRecCnt + 1 to 20 do
  ExecSQL; //SQL文を実行

SQL.Clear;

if Params.Count <> 0 then Params.Clear; //Paramコレクション

SQL.Add('UPDATE ' + lStrTblName + ' SET ');
SQL.Add('(');
SQL.Add('請求年, ');
SQL.Add('請求月, ');
SQL.Add('請求日, ');
```

行: 1 列: 1/19 文字: 1/19 コードページ: 932(shift_jis) RO Win

移行ツールの変換性能は・・？

ファイル名	読込件数	書込件数	自動変換				手変換				自 動 変換率
			変更 件数	挿入件数	削除件数	計	変更 件数	挿入件数	削除件数	計	
SMZP0240.dpr	42	44	0	2	0	2	0	0	0	0	100.0%
SMZP0250.dpr	42	44	0	2	0	2	0	0	0	0	100.0%
SMZP0260.dpr	42	44	0	2	0	2	0	0	0	0	100.0%
ufrmSMZP0240.dfm	1,619	1,619	1	0	0	1	0	0	0	0	100.0%
ufrmSMZP0250.dfm	1,071	1,071	1	0	0	1	0	0	0	0	100.0%
ufrmSMZP0260.dfm	1,586	1,586	1	0	0	1	0	0	0	0	100.0%
ufrmSMZP0240.pas	863	865	83	2	0	85	1	2	6	9	90.4%
ufrmSMZP0250.pas	754	756	50	2	0	52	1	2	9	12	81.3%
ufrmSMZP0260.pas	867	870	84	3	0	87	2	2	14	18	82.9%
urptSMZP0240.dfm	1,312	1,311	1	0	1	2	0	0	0	0	100.0%
urptSMZP0250.dfm	736	735	1	0	1	2	0	0	0	0	100.0%
urptSMZP0260.dfm	1,291	1,290	1	0	1	2	0	0	0	0	100.0%
urptSMZP0240.pas	70	70	3	0	0	3	0	0	0	0	100.0%
urptSMZP0250.pas	48	48	3	0	0	3	0	0	0	0	100.0%
urptSMZP0260.pas	69	69	3	0	0	3	0	0	0	0	100.0%
SMZP0270.dpr	42	44	1	2	0	3	0	0	0	0	100.0%
ufrmSMZP0270.dfm	1,239	1,239	1	0	0	1	0	0	0	0	100.0%
ufrmSMZP0270.pas	948	951	97	3	0	100	2	2	14	18	84.7%
ufrmSMZP0270.dfm	1,239	1,239	1	0	0	1	0	0	0	0	100.0%
urptSMZP0270.pas	58	58	3	0	0	3	0	0	0	0	100.0%
SMZP0280.dpr	42	44	1	2	0	3	0	0	0	0	100.0%
ufrmSMZP0280.dfm	1,214	1,214	1	0	0	1	0	0	0	0	100.0%
ufrmSMZP0280.pas	600	602	58	2	0	60	1	2	9	12	83.3%
urptSMZP0280.dfm	882	881	1	0	1	2	0	0	0	0	100.0%
urptSMZP0280.pas	54	54	3	0	0	3	0	0	0	0	100.0%

①

②

分類	自動変換率
入力系（①）	80～95%
帳票・バッチ（②）	100%

※ 自動変換率(%)=自動変換／(自動変換+手変換)

ツールを使って変換します

DEMONSTRATION

アジェンダ

- はじめに
- 移行の詳細
- ツールによる移行
- **移行後の対応**
- XenApp サーバーを使う場合の注意点
- まとめ

SQL文の修正 •• DECODE関数のエラーを修正

- BDEでは暗黙変換が行われていたが、FireDACではエラーになる

```
SQL.Add('SELECT ');
SQL.Add(' A.取引先コード ');
SQL.Add(',A.部課コード ');
 :
SQL.Add('A.雑区分, ');
SQL.Add('DECODE(A.雑区分,0,''通常'',1,''諸口'','' '') 雑区分名, ');
```

- CAST関数を使用して、明示的に変換・型のキャストを行うことで解消する

```
SQL.Add('SELECT ');
SQL.Add(' A.取引先コード ');
SQL.Add(',A.部課コード ');
 :
SQL.Add(',A.雑区分 ');
SQL.Add(',CAST(DECODE(A.雑区分,0,''通常'',1,''諸口'','' '') AS CHAR(4)) 雑区分名 ');
```

用紙サイズエラーへの対応

FUJI Xerox

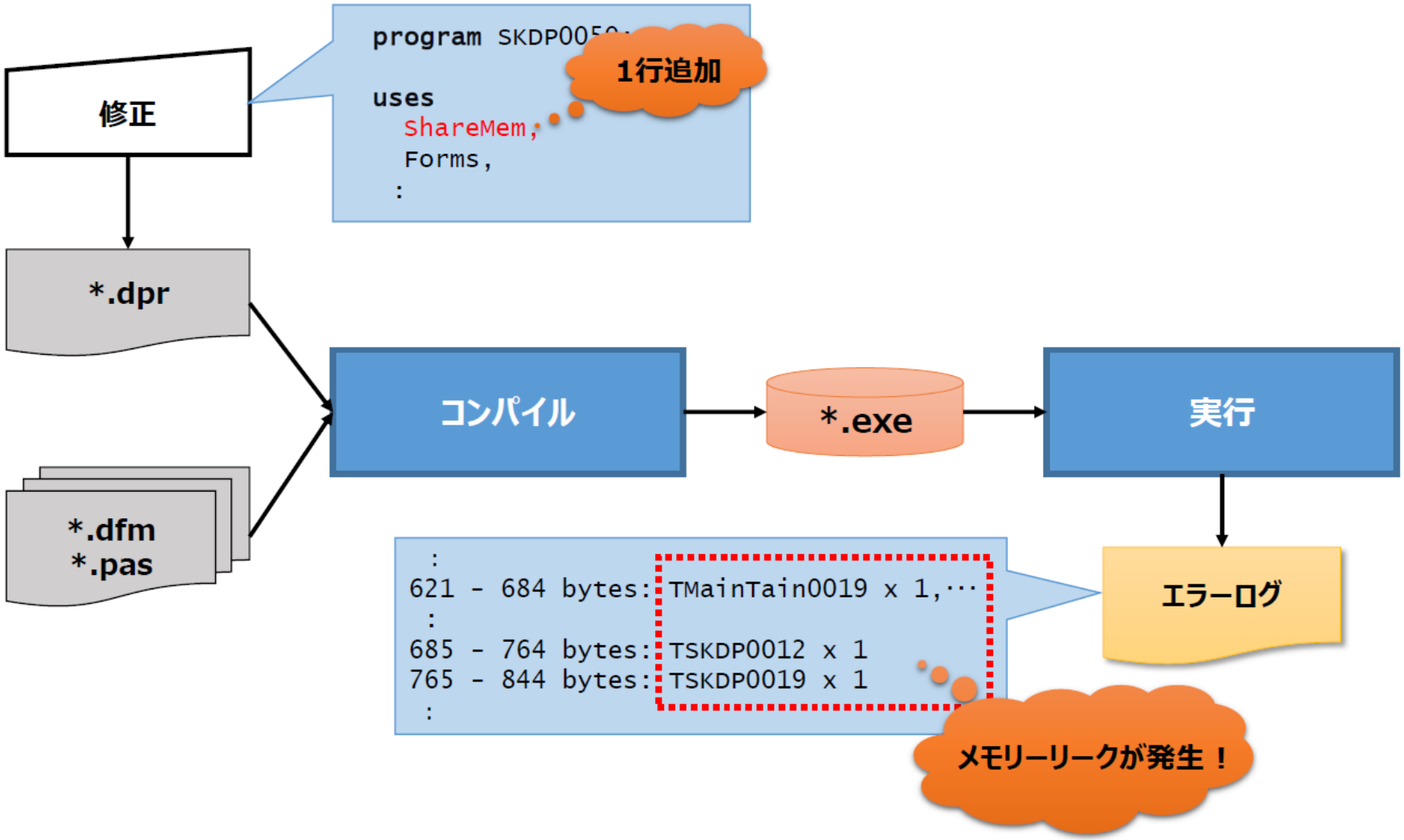
- Xerox社のDocuPrintC3360に印刷したとき、「016-799」エラーが発生する
- QuickReport 4.6 で定義されているB4用紙サイズがISOとJISで異なることが原因

```
unit QRPrntr;
  :
const
   cQRName = 'QuickReport 4.06'; { This string should not be resourced }
     :
   { Actual paper sizes for all the known paper types }
   cQRPaperSizeMetrics : array[Letter..A3ExtraTrans, 0..1] of extended =
      ((215.9, 279.4), { Letter }
          :
       (190.5, 254.0), { Executive }
       (297.0, 420.0), { A3 }
       (210.0, 297.0), { A4
       (210.0, 297.0), { A4 sma
       (148.0, 210.0), { A5 },
       (250.0, 354.0), { B4 }
       (182.0, 257.0), { B5 }
       (215.9, 330.2), { Folio }
```

B4用紙サイズ

```
(257.0, 364.0), { B4(JIS) }
```

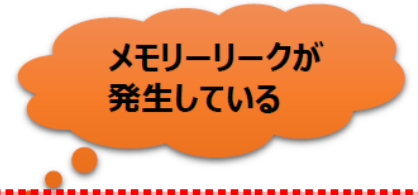
メモリーリークへの対応 … 調査の流れ



※詳細は、こちら http://edn.embarcadero.com/jp/article/33696

メモリーリークへの対応 … エラーログの内容（修正前）

```
  :
--------------------------------2016/5/10 14:03:12--------------------------------
This application has leaked memory. The small block leaks are:

5 - 12 bytes: String x 8
13 - 20 bytes: Unknown x 14
21 - 28 bytes: String x 7, Unknown x 4
29 - 36 bytes: Unknown x 1
237 - 268 bytes: TEraoperate x 2
621 - 684 bytes: TMainTain0019 x 1, TMainTain0018 x 1, TSKDP0018 x 1, TSKDP0017 x 1, TSKDP0016 x 1,
TSKDP0015 x 1, TSKDP0014 x 1, TMainTain0012 x 1
685 - 764 bytes: TSKDP0012 x 1
765 - 844 bytes: TSKDP0019 x 1

Note: This memory leak check is only performed if Delphi is currently running on the same computer.
Memory leak detail is logged to a text file in the same folder as this application. To disable this
memory leak check, undefine "EnableMemoryLeakReporting".
```



メモリーリークへの対応 … 廃棄されていないクラスを追加

```
procedure TfrmSKDP0050.lpClassListFree;
begin
  //入金伝票
  SKDP0010.Free;
  //出金伝票
  SKDP0011.Free;
  //支払手形出金伝票
  SKDP0013.Free;
  //リスト
  ClassList.Free;
end;
```

```
procedure TfrmSKDP0050.lpClassListFree;
begin
  //入金伝票
  SKDP0010.Free;
  //出金伝票
  SKDP0011.Free;
  //受取手形入金伝票
  SKDP0012.Free;
  //支払手形出金伝票
  SKDP0013.Free;
  //受取手形取立入金伝票
  SKDP0014.Free;
  //支払手形決済出金伝票
  SKDP0015.Free;
  //割引手形入金伝票
  SKDP0016.Free;
  //割引手形決済振替伝票
  SKDP0017.Free;
  //現預金振替伝票
  SKDP0018.Free;
  //一般振替伝票
  SKDP0019.Free;
  //クラスリスト
  ClassList.Free;
end;
```

追加

メモリーリークへの対応 … エラーログの内容（修正後）

```
  :
  :
--------------------------------2016/5/10 14:03:12--------------------------------
This application has leaked memory. The [illegible] are:

5 - 12 bytes: String x 4
13 - 20 bytes: Unknown x 4
237 - 268 bytes: TEraoperate x 1

Note: This memory leak check is only performed if Delphi is currently running on the same computer.
Memory leak detail is logged to a text file in the same folder as this application. To disable
this memory leak check, undefine "EnableMemoryLeakReporting".
```

メモリーリークが
解消されている

帳票作成時間を「劇的！」に削減する

・FireDACの「配列DML」機能を使う
・複数のDML(INSERT/UPDATE/DELETE)文をパラメータ付きで一度に実行
・高速実行の実現

[INSERTの例]

```
procedure TDataModule1.ExecuteDML;
var
  i: Integer;
begin
  // 実行するDML(INSERT)SQLをセット
  ADQuery1.SQL.Text:= 'INSERT INTO DML_TEST (TINT, TSTRING)
VALUES(:f1, :f2)';
  // 配列の大きさをセット
  ADQuery1.Params.ArraySize := 1000;
  // 配列に値をセット
  for i:=0 to 1000-1 do
  begin
    ADQuery1.Params[0].AsIntegers[i] := i;
    ADQuery1.Params[1].AsStrings[i]  := 'Str' + IntToStr(i);
  end;
  // SQLの実行
  ADQuery1.Execute(ADQuery1.Params.ArraySize, 0);
end;
```

メモリサイズに注意!

単位：ミリ秒

No.	帳票名	改善前	改善後
1	帳票A	129,756	1,548
2	帳票B	89,011	1,213
3	帳票C	158,812	1,578
4	帳票D	49,214	683
5	帳票E	28,150	506
6	帳票F	1,329	234
7	帳票G	1,334	151
8	帳票H	60,299	908

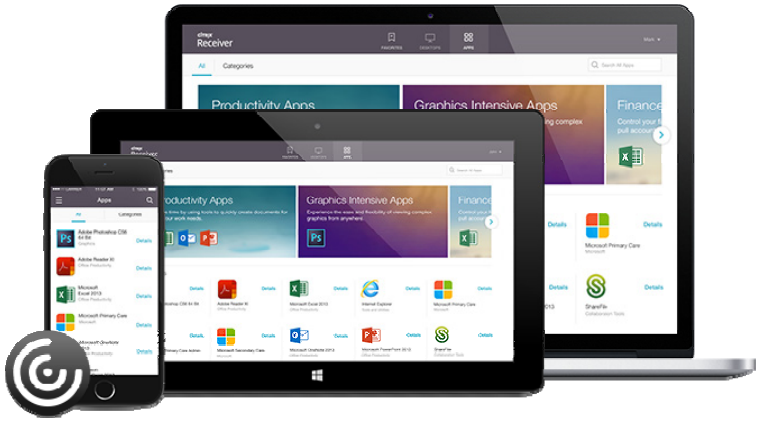
アジェンダ

- はじめに
- 移行の詳細
- ツールによる移行
- 移行後の対応
- **XenApp サーバーを使う場合の注意点**
- まとめ

なぜ、Citrix XenApp サーバーか?

メリットは・・

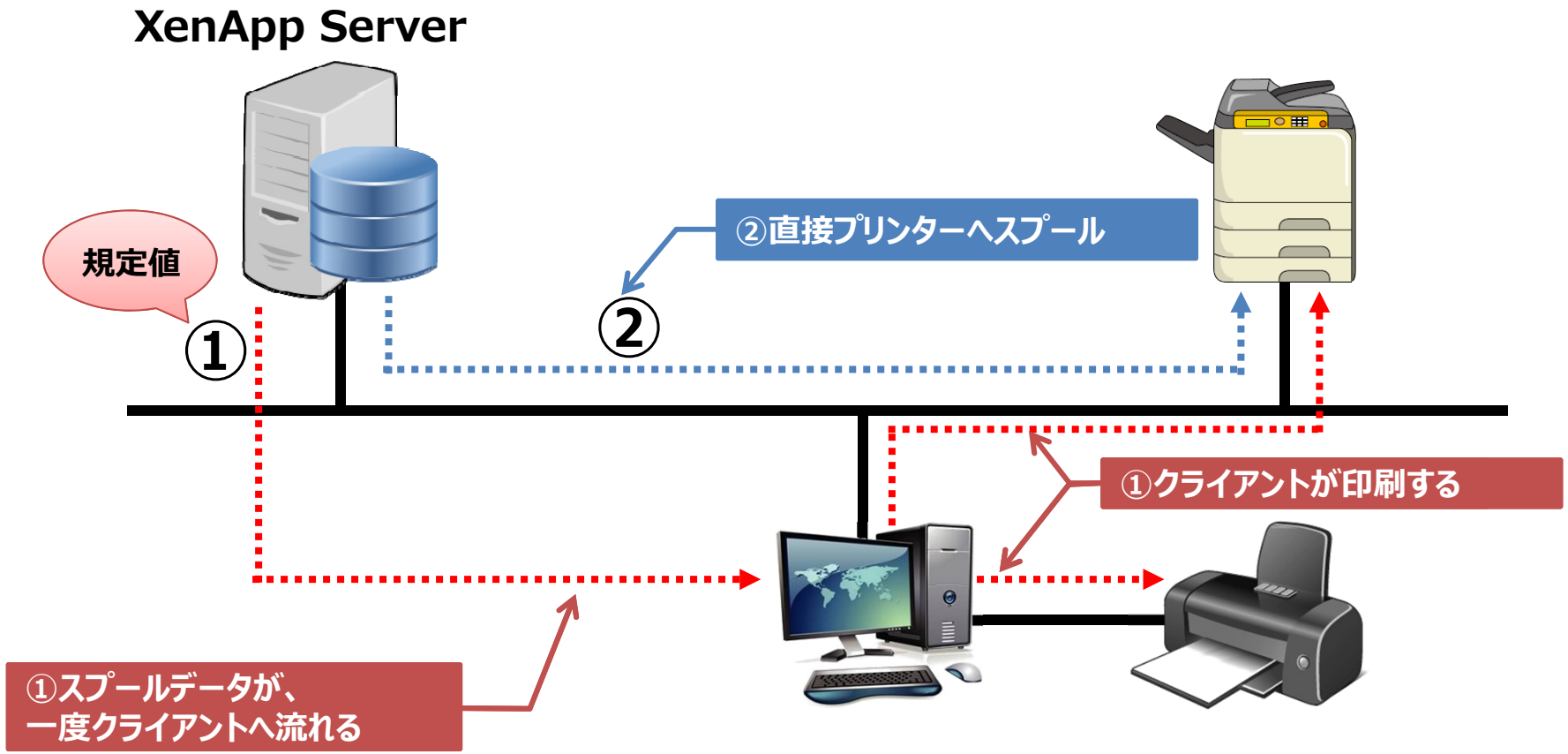
- **簡単にアプリケーションの仮想化ができる（アプリケーションをインストールするのはサーバーのみ）**
  テスト、インストール、アップデートの作業コストの削減できる
  業務システムだけではなくてOffice／Photoshop／CADも・・
- **リモートデスクトップ(Remote App)の1/20の帯域で使用することができる**
  狭帯域でも快適に動作するので外部からの接続でもストレスがない（ISDN程度あればよい）
- **負荷分散性能がいい**
  XenAppは標準で高性能な負荷分散が可能
  リモートデスクトップで負荷分散する場合、他のNLB(Network Load Balancing)製品の組合せが必要
- **クライアントOSの種類に関係なく展開することが可能**
  Windowsに限らず、iOSやAndroidの端末を使用することができる
  iOSで使用する場合でも「App Store」に載せる必要はない



ディメリットは・・

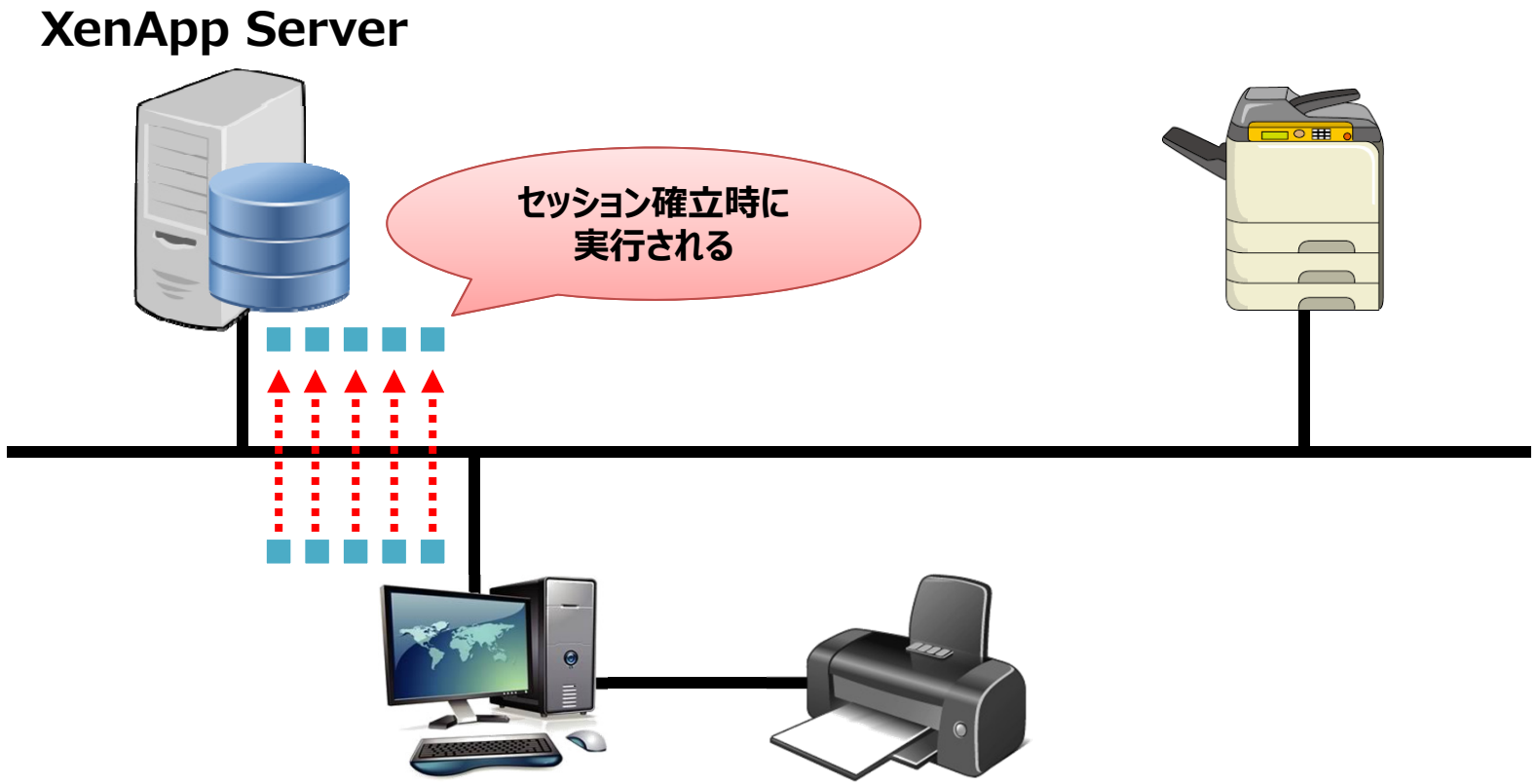
- **イニシャルコストが高い**
  リモートデスクトップのCALとXenAppのライセンスを購入する必要がある
- **XenAppを構築する知識と技術が必要**

XenAppでの印字データの流れ



※クライアントに流れるデータは、ネイティブドライバ(専用)とユニバーサル・プリンタ・ドライバ(汎用) (以下、UPDと略す)で異なる

XenAppでのクライアントプリンターのマッピング



※XenAppのポリシー設定により、ネイティブドライバーがインストールされていない場合、UPDを使用するか、マッピングから除外するか、を指定できる

マッピングでの注意事項

- **サーバー／クライアントのプリンタドライバを「最新」かつ「同一バージョン」にする**
  ※最新とは、プリンタメーカーから提供されているドライバの最新版という意味
  ※必ず**「Citrix対応状況」**を確認し、対応されているプリンターとドライバを使用する

- **メーカー推奨のユーティリティを利用する**
  ※Xeroxであれば「Print Utility for Citrix XenApp」
  http://www.fujixerox.co.jp/product/software/printutility_cxa/

- **マッピングには時間が掛かるため(デバイスで数秒) 、不要なドライバーは極力削除する**

- **アプリ側でプリンタ情報を取得する場合は、タイミングに注意が必要**
  ※マッピングが完了するまで「待機」する等の考慮が必要

- **ユニバーサルプリンター機能は、限定的な使用にとどめる**
  ※最近の複合機や罫線／イメージを多用している場合は、スプールデータが肥大化する

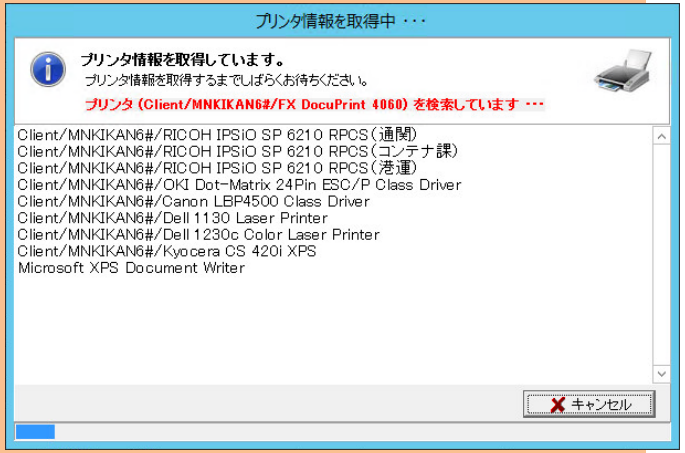
XenApp対応

- **出力プリンタの情報をIniファイルに記憶**
  帳票単位に、前回出力したプリンタの番号を記憶し、今回出力するプリンタの規定値が決定される

- **出力プリンタの「プリンタ名」をIniファイルに記憶**
  プリンタの番号から「プリンタの名」を記憶するよう仕様を変更した

- **QuickReportの内部処理をプリンタ番号から「プリンタ名」に変更（結構大変な処理）**
  VCL内部では、取得したプリンタ番号で（内部的にはTPrinterを使用）、出力プリンタが指定される
  しかし、XenAppは、前述したとおりマッピング動作が入るためQuickReportに直接手を入れ、プリンタ名で指定するように変更した

- **前回出力したプリンタがマッピングされるまで待つ処理を追加**
  印刷の冒頭に処理を追加し、マッピングを視覚的に分かるよう変更した



アジェンダ

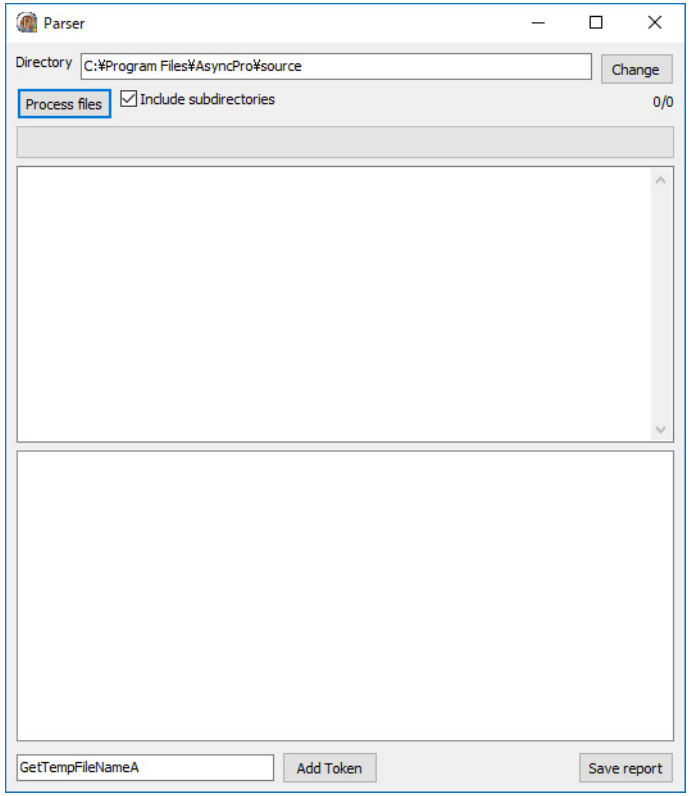
- はじめに
- 移行の詳細
- ツールによる移行
- 移行後の対応
- XenApp サーバーを使う場合の注意点
- **まとめ**

まとめ

- Delphiは、マイグレーションに最適！
- 移行ツールを過信せず、旧システムの調査を必ず実施すること
- 出現回数が多いものは、ツール化するメリットは大きい
- ツールですべて変換しようと思わない！

参考情報 (移行, Unicode)

- ✓ **SJISからUnicodeへ! マイグレーションテクニック**
  http://edn.embarcadero.com/article/images/40857/a4.pdf
- ✓ **Delphi2009ではじめるUnicodeアプリケーション -既存コード移行のポイント-**
  http://edn.embarcadero.com/jp/article/images/39112/a6.pdf?elqTrackId=d56d63716b3448e09d9949932339a978&elqaid=2506&elqat=2
- ✓ **旧Delphiで作成されたアプリケーションをDelphi2010に移行するには**
  http://edn.embarcadero.com/print/images/40857/a3.pdf
- ✓ **Delphiにおける一般的なUnicodeへの移行テクニック**
  https://www.embarcadero.com/images/jp/dm/technical-papers/delphi_unicode_wp_jp.pdf
- ✓ **Unicode Statistics Tool**
  http://cc.embarcadero.com/item/27398



参考情報 (BDE, FireDAC)

- ✓ **はじめてのFireDAC (PDF)**
  http://edn.embarcadero.com/jp/article/images/43368/a2.pdf

- ✓ **はじめてのFireDAC (動画)**
  https://www.youtube.com/watch?v=lDD6z4NuGEo
  https://www.youtube.com/watch?v=ZcLu6yq3wHE

- ✓ **BDEアプリを最新へ - FireDACによる移行**
  http://edn.embarcadero.com/print/images/43102/B2.pdf

- ✓ **BDE から FireDAC への移行 - Paradox から InterBase の場合**
  http://edn.embarcadero.com/jp/article/42974

- ✓ **FireDAC**
  http://docwiki.embarcadero.com/RADStudio/Berlin/ja/FireDAC

- ✓ **FireDAC への BDE アプリケーションの移行**
  http://docwiki.embarcadero.com/RADStudio/Berlin/ja/FireDAC_%E3%81%B8%E3%81%AE_BDE_%E3%82%A2%E3%83%97%E3%83%AA%E3%82%B1%E3%83%BC%E3%82%B7%E3%83%A7%E3%83%B3%E3%81%AE%E7%A7%BB%E8%A1%8C

- ✓ **データ型マッピング（FireDAC）**
  http://docwiki.embarcadero.com/RADStudio/Berlin/ja/%E3%83%87%E3%83%BC%E3%82%BF%E5%9E%8B%E3%83%9E%E3%83%83%E3%83%94%E3%83%B3%E3%82%B0%EF%BC%88FireDAC%EF%BC%89

- **SuperEDIT／SuperGRID のご購入は、こちらから**

株式会社オン・アンド・オン
http://o2components.on-and-on.biz/

株式会社コンポーネントソース
https://www.componentsource.co.jp/product/superedit-j
https://www.componentsource.co.jp/product/supergrid-j

- **システム移行のご相談は、こちらへ**
  o2c@on-and-on.biz

ご清聴
ありがとうございました

人と技術をつなぐ

Yoshiki.tanaka-avsoft@nifty.com