

【S4】テクニカルセッション

# 「現役ヘルプデスクが教える Delphi活用のポイント」

株式会社ミガロ. RAD事業部

課長

吉原 泰介



# アジェンダ

- ミガロ.について

## VCL for the WEBでWebサービス連携

### ■ VCL for the WEBでWebサービス連携

**補足資料1** XMLデータバインディングの使い方

## バージョンアップの移行ポイント

### ■ バージョンアップの事前調査とバージョン選定

### ■ バージョンアップ時によくある変更点

**補足資料2** 旧DelphiをWindows7へインストールする

# ミガロ.について

## 株式会社ミガロ. — A 100% IBM i Company —

### 会社情報

所在地: 本社 大阪市浪速区湊町2-1-57  
難波サンケイビル13F

東京事業所 東京都港区麻布台1-4-3  
エグゼクティブタワー麻布台11F

### 事業内容

IBM i 向けのソフトウェア・ツール販売および技術サポート

#### 【開発ツール】

- Delphi/400
- JC/400

#### 【バックアップ/災害対策ソフト】

- Maxava HA
- PlateSpin Forge

#### 【アプリケーション・ライフサイクル・マネージメント】

- MKS



- Delphi/400
- DelphiをIBM i (AS/400)に完全対応させたミドルウェア
- 国内約650社、全世界約5,000社の導入実績



# VCL for the WEBで Webサービス連携



# VCL for the WEBでWebサービス連携

## Webサービス連携実装例(郵便番号検索)

The screenshot shows a web browser window displaying a form for address search. The form includes fields for 'コード' (Code), '名称' (Name), '郵便番号' (Postal Code), '都道府県' (Prefecture), '市区町村' (City/Town/Village), and '町域等' (District/Block/etc.). A dropdown menu is open for the '郵便番号' field, showing a list of addresses. An orange arrow points from the '郵便番号' field to the dropdown. A blue box with the text '①郵便番号を入力' (1. Enter postal code) points to the '郵便番号' field. Another blue box with the text '②Webサービスより前方一致の住所リストを取得' (2. Retrieve list of addresses with forward matching from web service) points to the dropdown menu. The browser window has a Norton security bar and a Bing search bar. The status bar at the bottom indicates 'ページが表示されました' (Page displayed) and 'インターネット | 保護モード: 無効' (Internet | Protection mode: Off).

コード: 000001  
名称: 株式会社ミガロ.  
郵便番号: 556  
都道府県:   
市区町村:   
町域等:

5560006: 大阪府大阪市浪速区日本橋東  
5560004: 大阪府大阪市浪速区日本橋西  
5560017: 大阪府大阪市浪速区湊町  
5560016: 大阪府大阪市浪速区元町

①郵便番号を入力

②Webサービスより  
前方一致の住所リストを取得

保存 取消

地図表示

ページが表示されました

インターネット | 保護モード: 無効

# VCL for the WEBでWebサービス連携

## 今回利用する郵便番号検索Webサービス

グローブテクノロジー社提供Webサービス



<http://groovetechnology.co.jp/websearch/zipsearch/index.html>

仕様やインターフェースの詳細が公開されています。

- 3桁以上の郵便番号をHTTPを使って渡すとXMLの形で前方一致対象のデータを返却してくれます。

郵便番号:

556



5560006 : 大阪府大阪市浪速区

5560004 : 大阪府大阪市浪速区

サービス利用を明示  
する必要があるなど

Webサービスを利用場合、提供元利用規約をよく確認してください。



# VCL for the WEBでWebサービス連携

## 郵便番号検索サービス連携の仕組み

- 郵便番号(TIWEEdit)が入力されてExitする際に OnAysncExitイベントでWebサービスを利用して、結果をTIWListBoxにセットします。

556の郵便番号  
情報をリクエスト



TIdHTTP

TIdHTTPコンポーネントでリクエスト

郵便番号: 556

WEB SERVICE GROOVE Technology

5560006 : 大阪府大阪市浪速区日本橋東  
5560004 : 大阪府大阪市浪速区日本橋西  
5560017 : 大阪府大阪市浪速区湊町  
5560016 : 大阪府大阪市浪速区元町

結果  
XML

Webサービス  
(REST)



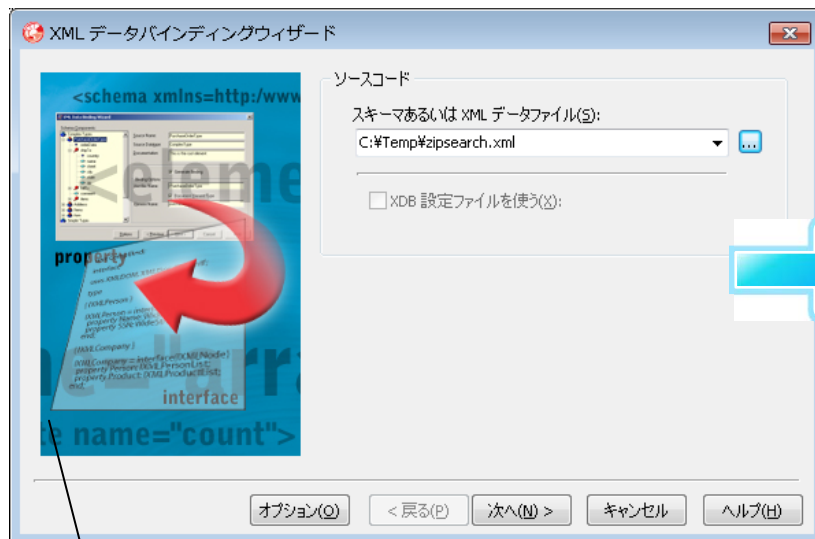
TXMLDocument

結果はTXMLDocumentで受け取り  
XMLデータバインディングで生成するクラスで処理する

# VCL for the WEBでWebサービス連携

## XMLデータバインディングで取得XMLのクラスを自動生成

- XMLデータバインディングで生成したユニットを、利用するプログラムで[ファイル|ユニットを使う]から指定するか、usesに追記します。



```
*****
XML データバインディング
作成日: 2009/12/01 00:00:00
作成元: Webサービスから得られるXMLデータ
設定ファイルの保管先: 同一フォルダ内(zipsearch.xdb)
*****
unit zipsearch;
interface
uses xmlidom, XMLDoc, XMLIntf;
type
{ 前方宣言 }

IXMLGroovewebServiceType = interface;
IXMLAddressType = interface;
IXMLAddressTypeList = interface;
IXMLOfficeType = interface;
IXMLOfficeTypeList = interface;

{ IXMLGroovewebServiceType }

IXMLGroovewebServiceType = interface(IXMLNode)
['{C156868D-BCCC-475B-A5F6-BD3C8606A230}']
{ プロパティ参照関数 }
function Get_Version: WideString;
function Get_Datetime: WideString;
function Get_Address: IXMLAddressTypeList;
function Get_Office: IXMLOfficeTypeList;
procedure Set_Version(Value: WideString);
procedure Set_Datetime(Value: WideString);
{ プロパティ }
property Version: WideString read Get_Version write Set_Version;
property Datetime: WideString read Get_Datetime write Set_Datetime;
```

使い方の詳細は補足資料を参照ください。



# VCL for the WEBでWebサービス連携

## 郵便番号検索サービスより受け取れるXMLデータ

- この郵便番号検索サービスでは次のような形でXMLが返却されます。データは生成されたソースのクラスで簡単に処理できます。

フィールド	説明
address	住所の郵便番号の情報です。zipcode, prefecture, city, town, prefecture_yomi, city_yomi, town_yomiが設定されます。
office	事業所の個別郵便番号の情報です。zipcode, prefecture, city, town, street, office_name, office_name_yomiが設定されます。
zipcode	郵便番号が、「9999999」の形式で設定されます。
prefecture	都道府県名が設定されます。
city	市区町村名が設定されます。
town	町域名が設定されます。
office_name	大口事業所等名が設定されます。
street	大口事業所等名の小字名、丁目、番地等が設定されます。
prefecture_yomi	都道府県名のよみ(カタカナ)が設定されます。
city_yomi	市区町村名のよみ(カタカナ)が設定されます。
town_yomi	町域名のよみ(カタカナ)が設定されます。
office_name_yomi	大口事業所等名のよみ(カタカナ)が設定されます。

# VCL for the WEBでWebサービス連携

## 郵便番号検索サービスのリクエストURL

- 今回利用する郵便番号検索サービスはURLにパラメータを付与してアクセスするとXMLの形で結果を得ることができます。

556の郵便番号を検索する場合は次のようなURLになります。

共通アドレス部分

[http://groovetechnology.co.jp/ZipSearchService/v1/zipsearch  
? format=xml&ie=UTF8&oe=UTF8&zipcode=556](http://groovetechnology.co.jp/ZipSearchService/v1/zipsearch?format=xml&ie=UTF8&oe=UTF8&zipcode=556)

?以降がパラメータ部分  
Format:返却形式  
ie:インプット文字コード  
oe:アウトプット文字コード

zipcodeというパラメータで郵便番号を指定。ここにIWEditの入力値をセットする。

# VCL for the WEBでWebサービス連携

## 住所データ取得(ソース)

```
procedure TfrmWebSample2Input.edtMCYUBNAsyncExit(Sender: TObject;
  EventParams: TStringList);
var
  i: Integer;
  URI: String;
  RootNode: IXMLGroovewebseviceType; //XMLデータバインディングで取り込んだクラス
begin
  // リクエストするURLをセット
  URI := 'http://groovetechnology.co.jp/ZipSearchService/v1/zipsearch?'
    + 'format=xml&ie=UTF-8&oe=UTF-8&zipcode=' + iedtMCYUBN.Text;
  // レスポンスとして戻されたXMLを処理する
  XMLDocument1.XML.Text := IdHTTP1.Get(URI);
  RootNode := Getgroovewebsevice(XMLDocument1);
  lstMCYUBN.Items.Clear; // 住所リストをクリア
  // 受け取ったXMLデータから住所リストを作成
  for i := 0 to RootNode.Address.Count - 1 do
    lstMCYUBN.Items.Add(Format('%d : %s%s%s',
      [RootNode.Address.Items[i].Zipcode, //郵便番号
      RootNode.Address.Items[i].Prefecture, //都道府県名
      RootNode.Address.Items[i].City, //市町村名
      RootNode.Address.Items[i].Town])); //町域名
  // 住所リストが1件でもあれば先頭データを選択状態に
  if lstMCYUBN.Items.Count > 0 then
    lstMCYUBN.ItemIndex := 0;
end;
```

### リクエストURL編集

### ソース内コンポーネント名

- 郵便番号入力  
iedtMCYUBN : TIWEdit
- 住所リスト表示  
lstMCYUBN : TIWListbox
- 処理用  
XMLDocument1 : TXMLDocument  
IdHTTP1 : TIdHTTP

# VCL for the WEBでWebサービス連携

## カスタマイズ例

コード: 000001

名称: 株式会社ミガロ.

郵便番号: 5560017

WEB SERVICE GROOVE  
technology

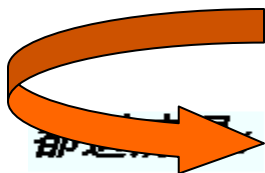
①ダブルクリックすると  
画面項目にセット

5560006 : 大阪府大阪市浪速区日本橋東

5560004 : 大阪府大阪市浪速区日本橋西

5560017 : 大阪府大阪市浪速区湊町

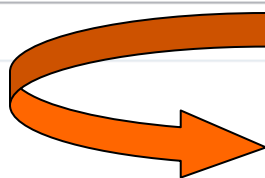
5560016 : 大阪府大阪市浪速区元町



都道府県: 大阪府

市区町村: 大阪市浪速区

町域等: 湊町



保存

取消

### 地図表示



②住所からさらにWebサービスを利用して地図情報を表示など

# VCL for the WEBでWebサービス連携

## Client/Server型のアプリケーションで利用

- Client/Server型のアプリケーションでもデータバインディングで取り込んだソースとTIdHttp、TXmlDocumentで実装の方法はほぼ同じです。

取引先データ入力

コード 000001

名称 株式会社ミガロ.

郵便番号 556

5560005 : 大阪府大阪市浪速区日本橋  
5560006 : 大阪府大阪市浪速区日本橋東  
5560004 : 大阪府大阪市浪速区日本橋西  
**5560017 : 大阪府大阪市浪速区湊町**  
5560016 : 大阪府大阪市浪速区元町

都道府県 地図表示

市町村

町域等

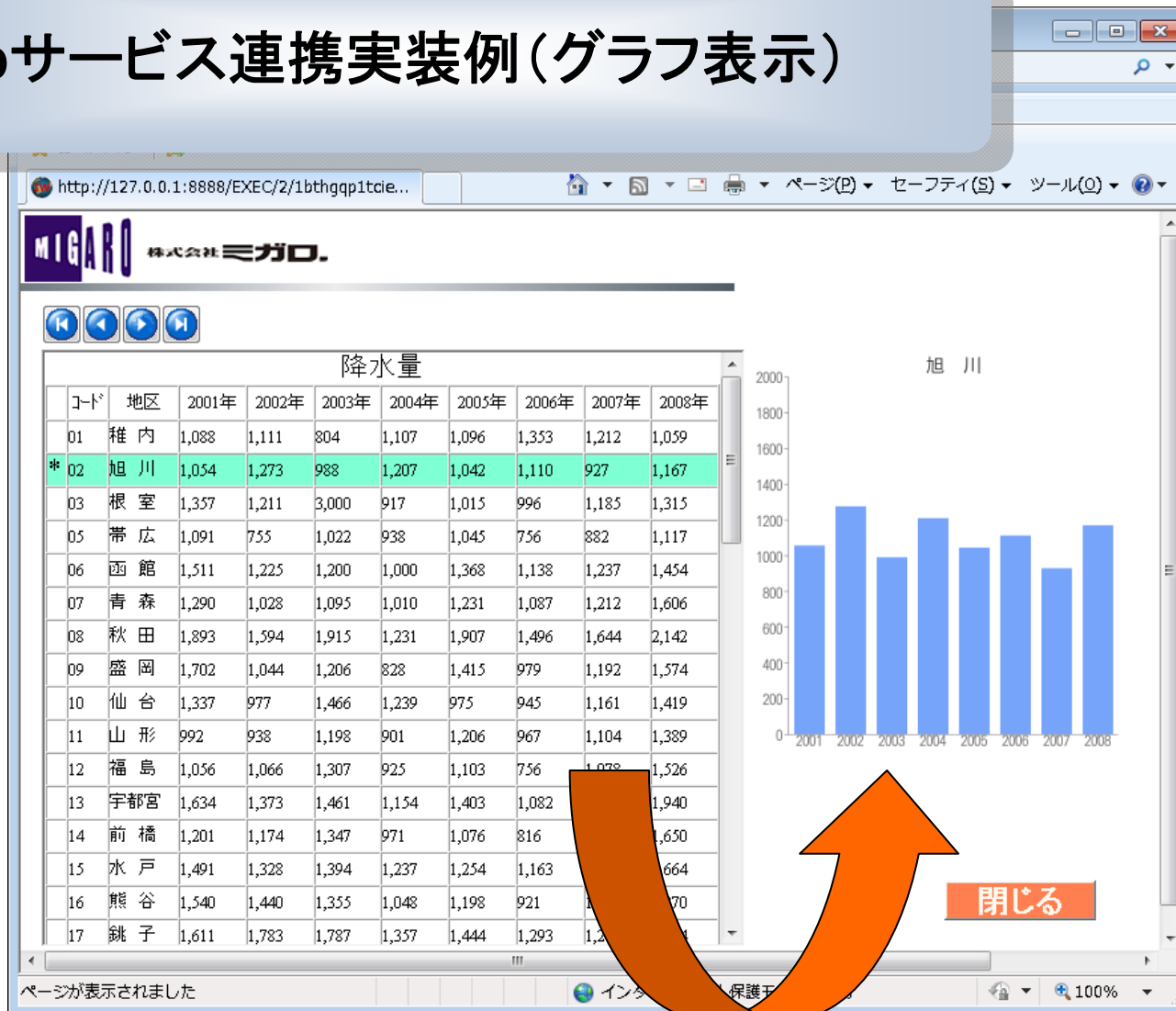
閉じる(C)

**Webサービスを利用場合、提供元利用規約をよく確認してください。**



# VCL for the WEBでWebサービス連携

## Webサービス連携実装例(グラフ表示)



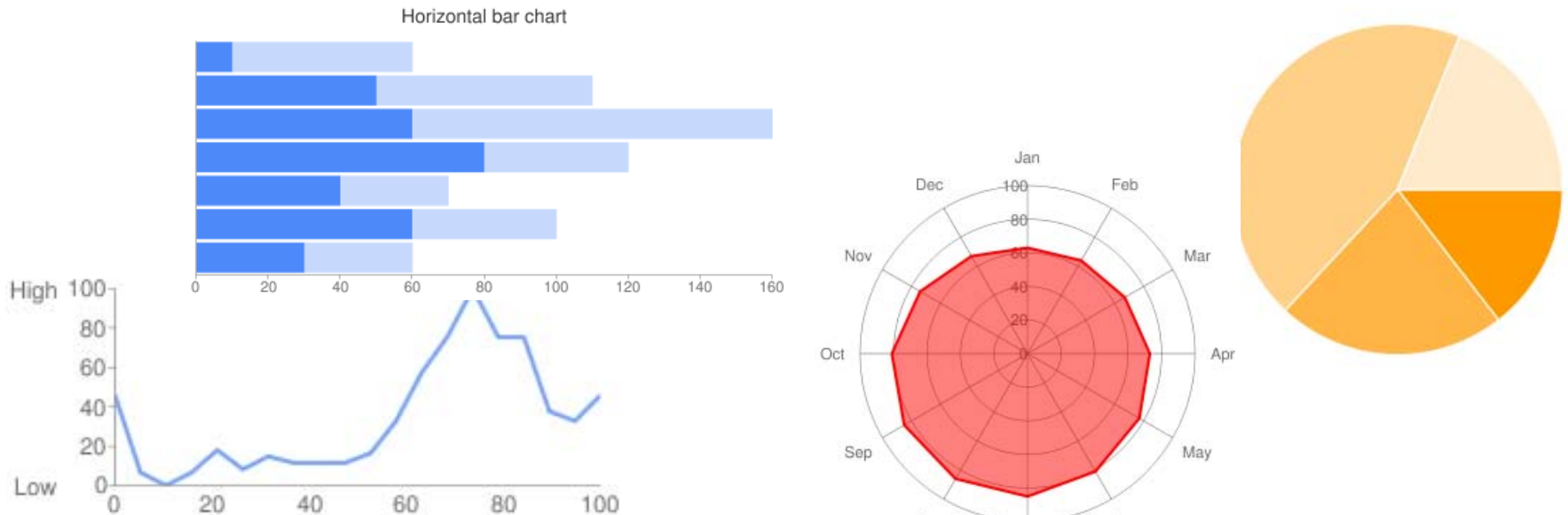
# VCL for the WEBでWebサービス連携

## 今回利用するグラフ生成Webサービス

GoogleのWebサービスの1つGoogle Chart API を利用する

<http://code.google.com/intl/ja/apis/chart/index.html>

インターフェースパラメータ仕様やサンプルなどが公開されています。



Webサービスを利用場合、提供元利用規約をよく確認してください。

# VCL for the WEBでWebサービス連携

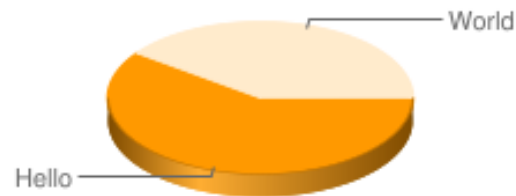
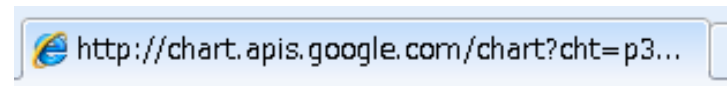
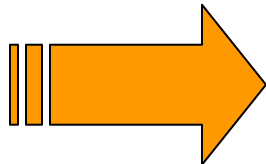
## Google Chart API でのグラフ生成

- 今回グラフに使用するGoogle Chart API はURLにパラメータを付与してアクセスすると簡単に結果を得ることができます。

次のようなURLでアクセスすると

<http://chart.apis.google.com/chart?cht=p3&chd=t:60,40&chs=250x100&chl=Hello|World>

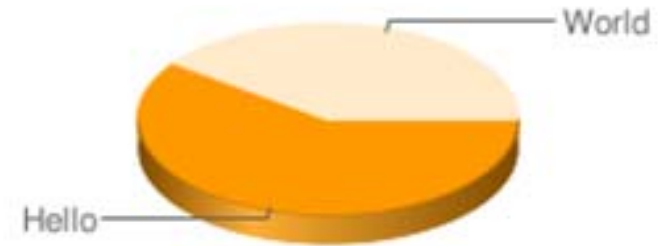
パラメータ通りのグラフが  
作成できます。



# VCL for the WEBでWebサービス連携

## Google Chart APIのリクエストURL

- リクエストURLの構造例<パラメータと値>
  - ・共通アドレス: <http://chart.apis.google.com/chart?>
  - ・サイズ: [chs=500x200](#)
  - ・割合: [chd=t:60,40](#)
  - ・グラフの種類: [cht=p3](#)
  - ・ラベル: [chl=Hello|World](#)



呼び出し構造は

共通アドレス?(パラメータ1)&(パラメータn)...

呼び出しアドレスに続いてパラメータを付け足していく

# VCL for the WEBでWebサービス連携

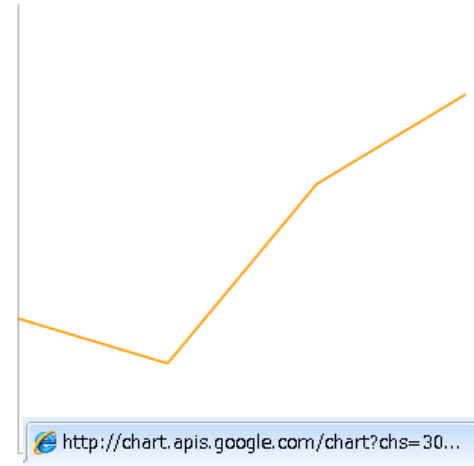
## Google Chart APIのリクエストURL

- パラメータでグラフの種類を変える。  
cht(グラフの種類)が違うと

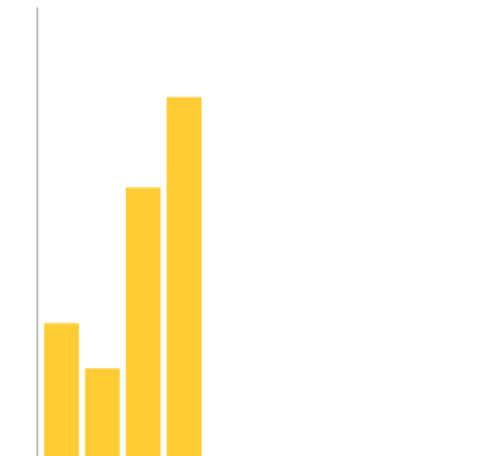
[http://chart.apis.google.com/chart?chs=300x300  
&chd=t:30,20,60,80&cht=lc](http://chart.apis.google.com/chart?chs=300x300&chd=t:30,20,60,80&cht=lc)



<http://chart.apis.google.com/chart?chs=300x300&chd=t:30,20,60,80&cht=lc>



[http://chart.apis.google.com/chart?chs=300x300  
&chd=t:30,20,60,80&cht=bvs](http://chart.apis.google.com/chart?chs=300x300&chd=t:30,20,60,80&cht=bvs)

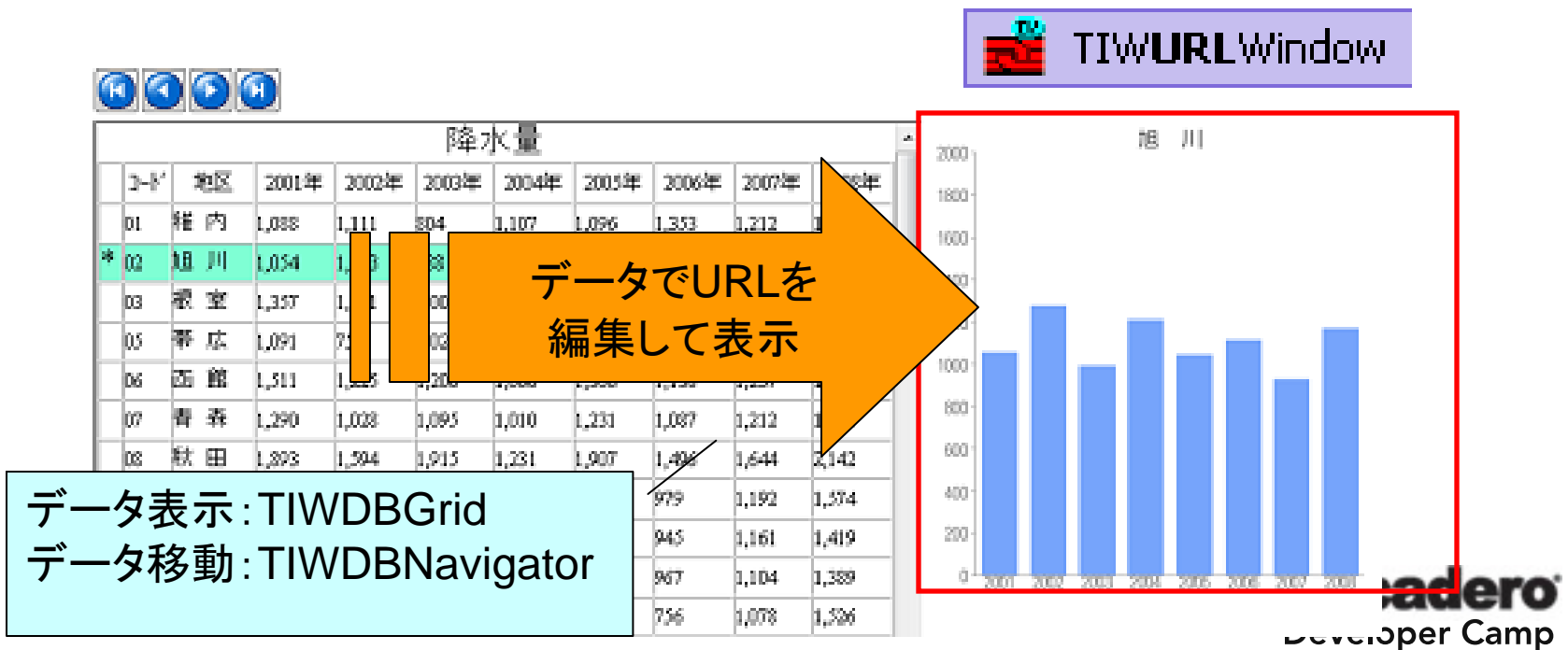




# VCL for the WEBでWebサービス連携

## Google Chart APIサービス連携の仕組み

- Delphi側でグラフに表示したいデータやグラフの種類に基づいたパラメータでURLを作成してTIWURLWindowコンポーネントで表示する。



# VCL for the WEBでWebサービス連携

## BarChart(棒グラフ)のパラメータ

- 今回使用するBarChart(棒グラフ)のパラメータは次の通り

■タイトル(エンコード文字-UTF8)

chtt=%E7%A8%9A%E3%80%80%E5%86%85

■グラフの種類(棒グラフ)

cht=bvg

■サイズ

chs=300x300

■色

chco=76A4FB

■ラベル表示

chxt=x,y

■Xラベル

chxl=0:|2001|2002|2003|....

■Y目盛

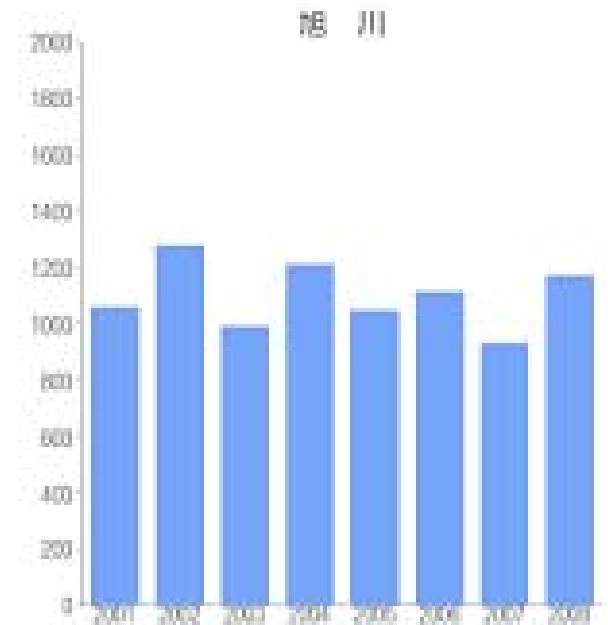
chxr=1,0,2000

■最小/最大値

chds=0,2000

■データ

chd=t:1100,1200,1300,....



※他にも使用できるパラメータは多数あります。

# VCL for the WEBでWebサービス連携

## グラフURLの作成手続き実装(ソース)

```
//グラフURL作成手続き
procedure TSampleChartfrm. CrtChart;
var
    //パラメータ編集用
    sURL, sChxl, sChxt, sChs, sChds, sChxr, sCht, sChco, sChd, sChtt: String;
begin
    //共通URL
    sURL := 'http://chart.apis.google.com/chart';
    //タイトル
    sChtt := '?cht=' + HttpEncode(UTF8Encode(ClientDataSet1.FieldByName('地区').AsString));
    //グラフ種類
    sCht := '&cht=bvg';
    //サイズ
    sChs := '&chs=300x300';
    //色
    sChco := '&chco=76A4FB';
    //表示ラベル
    sChxt := '&chxt=x,y';
    //xラベル
    sChxl := '&chxl=0:|2001|2002|2003|2004|2005|2006|2007|2008';
    //y目盛
    sChxr := '&chxr=1,0,2000';
```

ソース内コンポーネント名

■TIWDBGrid表示データセット

ClientDataSet1: TClientDataSet

■フィールド

地区: 地区名

200X年: 年度毎降水量

■グラフ表示

IWURLWindow1: TIWURLWindow

タイトルはデータ(ここでは地区というフィールド)からセットする。  
ChartのタイトルはUTF8なので  
UTF8EncodeしてからHttpEncodeで  
“旭 川”なら %E7%A8%9A%E3%80%80%E5%86%85  
のようにセットされる。

# VCL for the WEBでWebサービス連携

## グラフURLの作成手続き実装(ソース)

```
//データ域
sChds := '&chds=0, 2000';
//データ
sChd := '&chd=t:'
      + ClientDataSet1.FieldByName('2001年').AsString + ','
      + ClientDataSet1.FieldByName('2002年').AsString + ','
      + ClientDataSet1.FieldByName('2003年').AsString + ','
      + ClientDataSet1.FieldByName('2004年').AsString + ','
      + ClientDataSet1.FieldByName('2005年').AsString + ','
      + ClientDataSet1.FieldByName('2006年').AsString + ','
      + ClientDataSet1.FieldByName('2007年').AsString + ','
      + ClientDataSet1.FieldByName('2008年').AsString;
//URLを結合して設定
IWURLWindow1.URI := sURL + sChtt + sCht + sChs + sChco + sChxt + sChxl + sChxr + sChds + sChd;
end;
```

データはデータセットよりカンマ編集で結合してセットする。  
(ここでは200X年という8フィールド固定)  
フィールドではなく、レコードなどで数が可変ならラベルとあわせてループロジックで編集する。

# VCL for the WEBでWebサービス連携

## 初期処理、データ移動時に手続き呼出(ソース)

```
//初期処理
procedure TSampleChartfrm. IWebFormCreate(Sender: TObject);
begin
    //データOpen
    ClientDataSet1.Open;
    //グラフ作成
    CrtChart;
end;

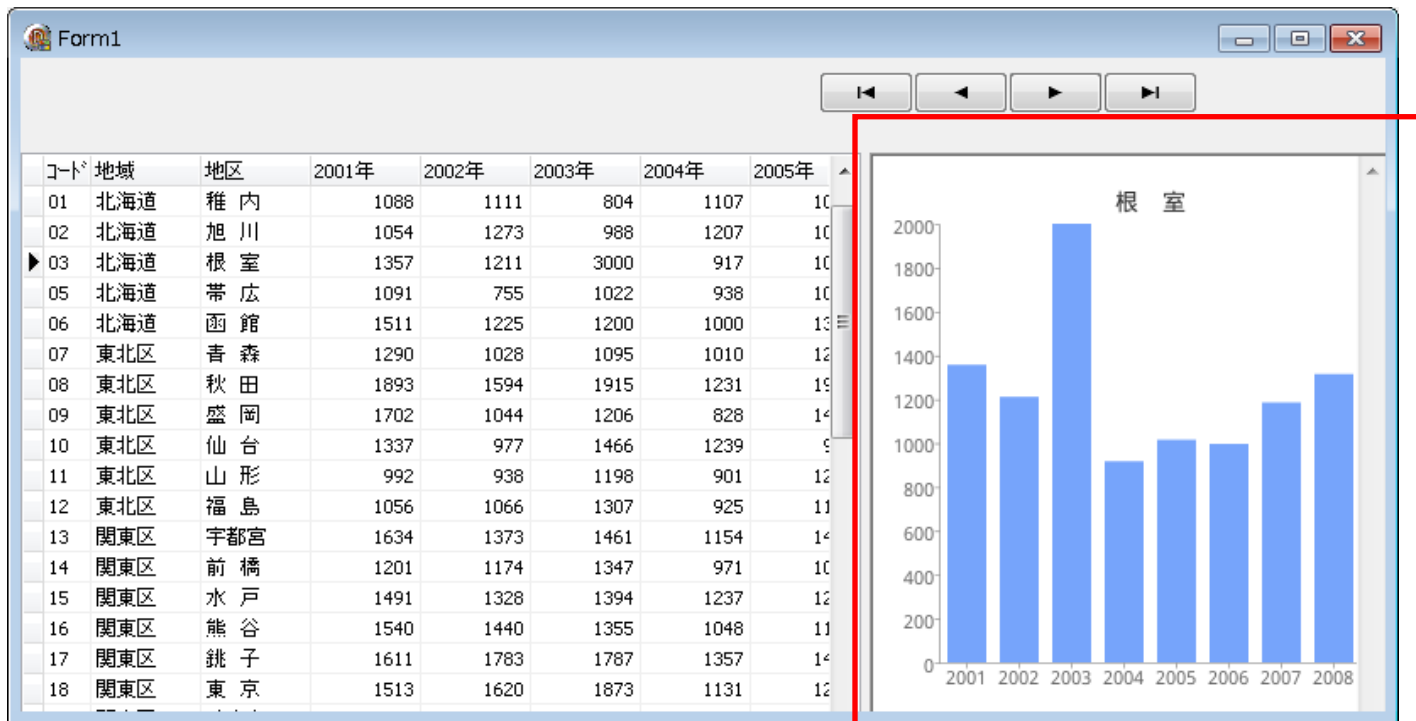
//データセットのAfterScrollイベントでデータ移動時にグラフ更新
procedure TSampleChartfrm. ClientDataSet1AfterScroll(DataSet: TDataSet);
begin
    //グラフ作成
    CrtChart;
end;
```



# VCL for the WEBでWebサービス連携

## Client/Server型のアプリケーションで利用

- Client/Server型のアプリケーションでもTIWURLWindowの代わりにTWebBrowserコンポーネントを経由して表示すれば使うこともできます。



Webサービスを利用場合、提供元利用規約をよく確認してください。



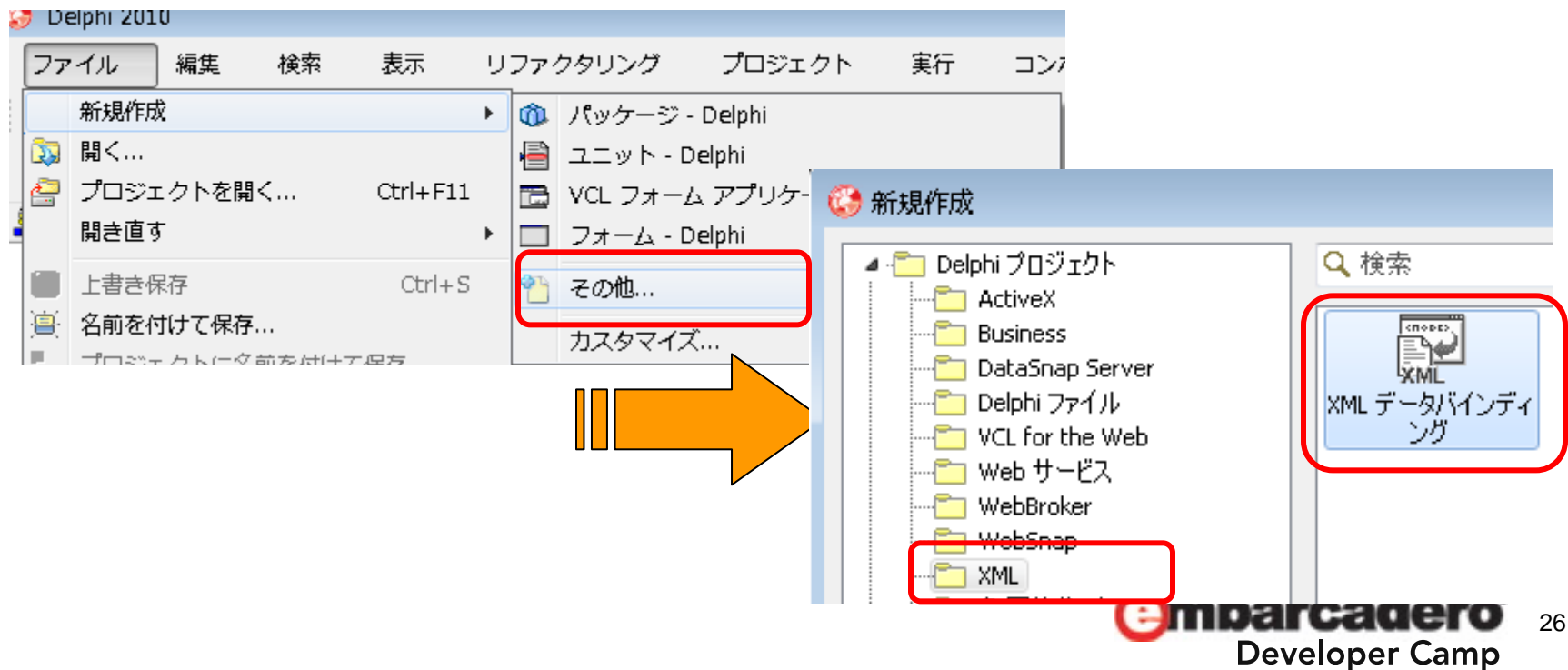
# 補足資料1

## XMLデータバインディングの 使い方



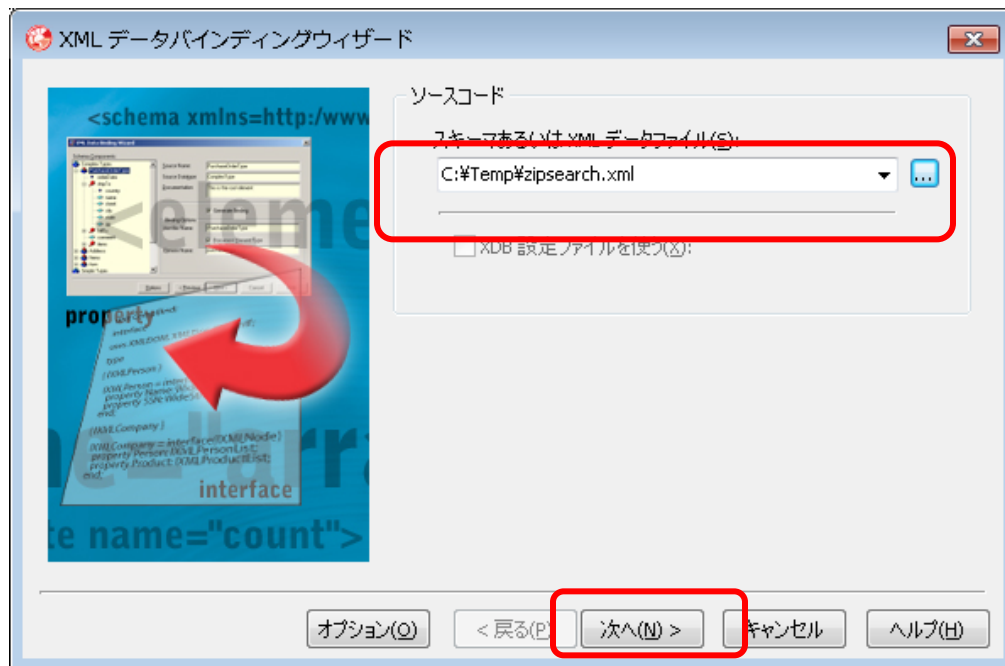
## XMLデータバインディングの利用①

- プロジェクトを開いたまま[ファイル|新規作成|その他]からXMLカテゴリでXMLデータバインディングを選択



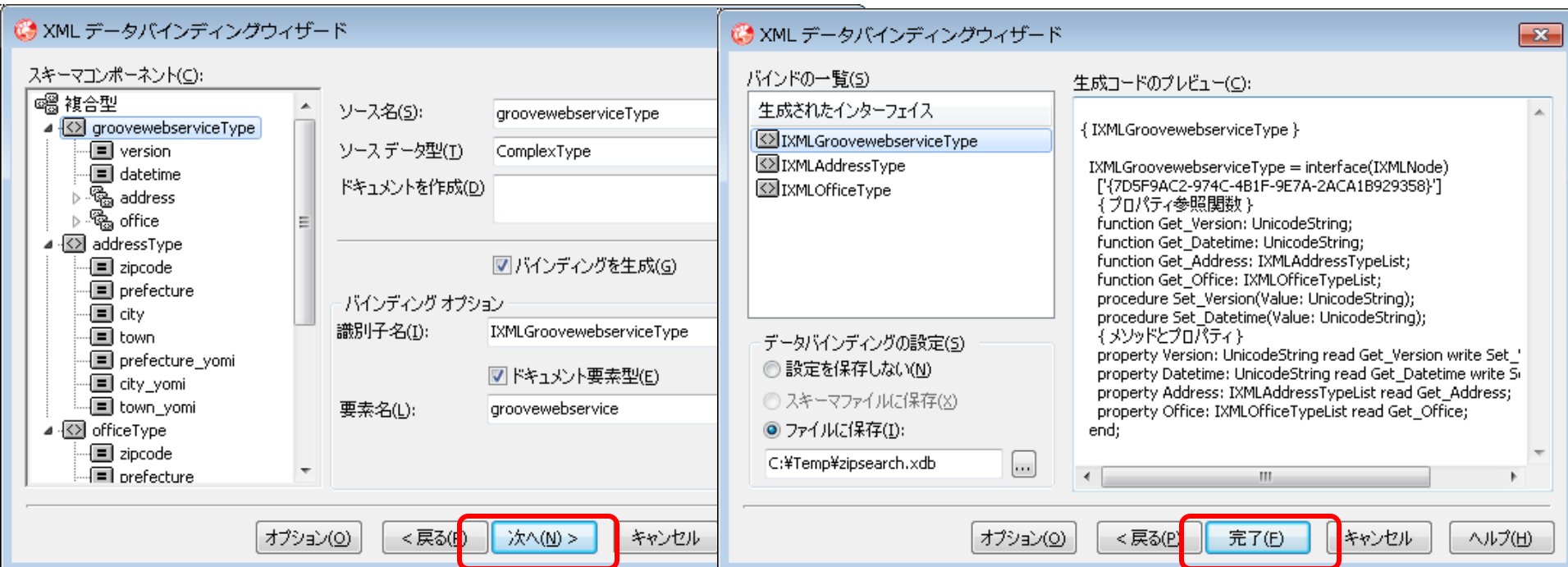
### XMLデータバインディングの利用②

- Webサービスで返却されるXMLサンプルを保存しておき、XMLを指定。  
(スキーマファイルがある場合はスキーマファイルで可能)



### XMLデータバインディングの利用③

- 解析結果に従いコードがプレビューされるので、ウィザードを進めて完了するとソースコードが生成されます。





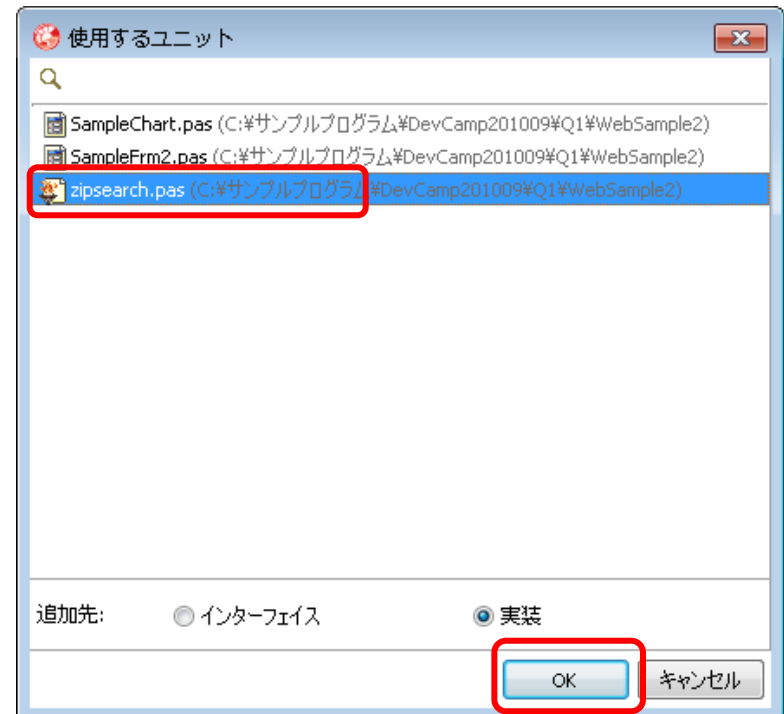
### XMLデータバインディングの利用④

- Delphiのソースが生成されたら、利用したいプログラムで  
[ファイル|ユニットを使う]から指定するか、usesに追記します。

```
*****
XML データバインディング
作成日: 2009/12/01 00:00:00
作成元: Webサービスから得られるXMLデータ
設定ファイルの保管先: 同一フォルダ内(zipsearch.xdb)
*****
unit zipsearch;

interface
uses xmlDom, XMLDoc, XMLIntf;
type
{ 前方宣言 }
IXMLGroovewebserviceType = inter
IXMLAddressType = interface;
IXMLAddressTypeList = interface;
IXMLOfficeType = interface;
IXMLOfficeTypeList = interface;
{ IXMLGroovewebserviceType }
IXMLGroovewebserviceType = interface(IXMLNode)
[ '{C156868D-BCCC-475B-A5F6-BD3C8606A230}' ]
{ プロパティ参照関数 }
function Get_Version: WideString;
function Get_Datetime: WideString;
function Get_Address: IXMLAddressTypeList;
function Get_Office: IXMLOfficeTypeList;
procedure Set_Version(Value: WideString);
procedure Set_Datetime(Value: WideString);
{ メソッドとプロパティ }
property Version: WideString read Get_Version write Set_Version;
property Datetime: WideString read Get_Datetime write Set_Datetime;
*****
```

生成されたソース





## バージョンアップの移行ポイント



# バージョンアップの移行ポイント

- バージョンアップの事前調査とバージョン選定
- バージョンアップ時によくある変更点
- バージョンアップ関連情報

# バージョンアップの移行ポイント

## バージョンアップの事前調査

- 確認すべき注意ポイント
  - － コンポーネント、サードパーティ、ミドルウェア製品
    - 使用しているコンポーネントや製品がバージョンアップに対応しているか。  
※ソース付属のみになっているコンポーネントもあるので注意。
    - フリーコンポーネントであればソースがあるか。
    - 代替コンポーネントや製品はあるか。
  - － レポートツール
    - QuickReportを使用している場合、新しいバージョンでは付属しない。  
RaveReports 等別製品へ代替移行するか。正式製品版を購入するか。
  - － データベース
    - BDE、DBexpress、ADO・・・手法切替が必要か。  
→ 手法を切替えるなら規模/工数。
    - DBのバージョンも含め、適切なドライバが存在するか。

# バージョンアップの移行ポイント

## バージョンアップのバージョン選定

- Delphiバージョンによる違い

Delphiバージョン		1	2	3	4	5	6	7	2005	2006	2007	2009	2010	XE
動作環境	Windows3.1	○	○	○	-	-	-	-	-	-	-	-	-	-
	WindowsNT3.51	-	○	○	-	-	-	-	-	-	-	-	-	-
	Windows95	-	○	○	○	○	-	-	-	-	-	-	-	-
	WindowsNT4	-	-	○	○	○	○	-	-	-	-	-	-	-
	Windows98	-	-	-	○	○	○	○	-	-	-	-	-	-
	WindowsMe	-	-	-	-	-	○	-	-	-	-	-	-	-
	Windows2000	-	-	-	-	-	○	○	○	○	○	○	○	○
	WindowsXP	-	-	-	-	-	-	○	○	○	○	○	○	○
	WindowsServer2003	-	-	-	-	-	-	-	○	○	○	○	○	○
	WindowsServer2008	-	-	-	-	-	-	-	-	-	-	○	○	○
	Windows Vista	-	-	-	-	-	-	-	-	-	○	○	○	○
	Windows 7	-	-	-	-	-	-	-	-	-	-	-	○	○
文字コード		Ansi	Ansi	Ansi	Ansi	Ansi	Ansi	Ansi	Ansi	Ansi	Ansi	Unicode	Unicode	Unicode
データベース	BDE	○	○	○	○	○	○	△	△	△	△	△	△	△
	ADO	-	-	-	-	○	○	○	○	○	○	○	○	○
	dbExpress	-	-	-	-	-	○	○	○	○	○	○	○	○
帳票ツール	QuickReport	-	○	○	○	○	○	○	-	無償DL	無償DL	-	-	-
	RaveReports	-	-	-	-	-	-	○	○	○	○	○	○	○
インストーラ 作成ツール	InstallShield	-	○	○	○	○	○	○	○	○	-	-	-	-
	InstallAware	-	-	-	-	-	-	-	-	-	○	○	○	○

※SP等の詳細情報は割愛

# バージョンアップの移行ポイント

## バージョンアップのバージョン選定

### ポイント



- Delphi2007以前とDelphi2009以降がバージョンの大きな境界線
  - AnsiからUnicodeへ変更されているので、リコンパイルだけでは難しい。

### ポイント



- Delphi2007がAnsi版の最終バージョン
  - 移行作業に工数がかけられず、Unicodeに対応する必要性がないのであれば旧アプリケーションはDelphi2007でバージョンを止めるのも選択肢。  
Delphi2009以降で無理なAnsi加工対応より◎

### ポイント



- Delphi2006以前から最新版へのバージョンアップする場合
  - ①一度Delphi2007にバージョンアップ
  - ②最新版へバージョンアップの段階バージョンアップを推奨
  - バージョン移行と文字コードの問題を混在させると・・・余計に大変になる場合も多い。



# バージョンアップの移行ポイント

## バージョンアップのバージョン選定

- Delphi2007を持っていない場合、最新のDelphiXEを購入すると入手することが可能です。  
(Delphi7 ,Delphi2007, Delphi2009, Delphi2010のダウンロード+ シリアルを入手できます。**製品入手は180日間以内**)

## 旧バージョンのライセンスの取得方法について

<http://edn.embarcadero.com/article/40874>

### ■注意

バージョンアップでXEを入手した場合は、既に所持している旧バージョンのシリアルを新たに入手することはできません。

# バージョンアップの移行ポイント

## バージョンアップのバージョン選定

### ポイント



- BDEを使用している場合、変更するか継続するか。

### BDEを継続する場合

#### メリット

- 変更の工数がかからない。

#### デメリット

- 製品の開発、サポートが終了しているので、使用する場合は開発者の自己責任が伴う。
- リモートデータベースに対するSQLリンクドライバは付属していない。  
(製品外で提供されるドライバは提供元次第)

# バージョンアップの移行ポイント

## バージョンアップのバージョン選定

### ポイント



- BDEを使用している場合、変更するか継続するか。

### dbExpress等に変更する場合

#### メリット

- 製品サポートの範囲で開発できる。(BDEより将来性が○)
- 配布環境でBDE自体のインストールが不要になる。  
(特にVistaや7でのUACの考慮が手間になる)

#### デメリット

- プログラムの作りや規模によっては膨大な工数がかかる。  
※双方向データセットの動作については、ClientDataSetを経由すればBDEのデータセットに近い動作は可能。

# バージョンアップの移行ポイント

## バージョンアップのバージョン選定

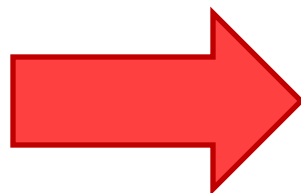
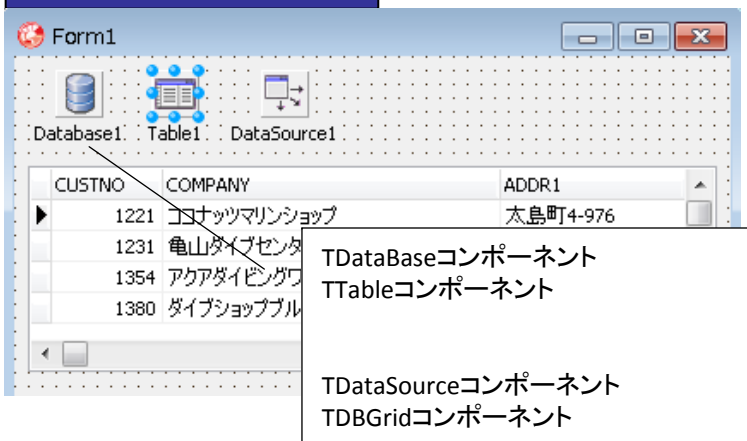
### ポイント



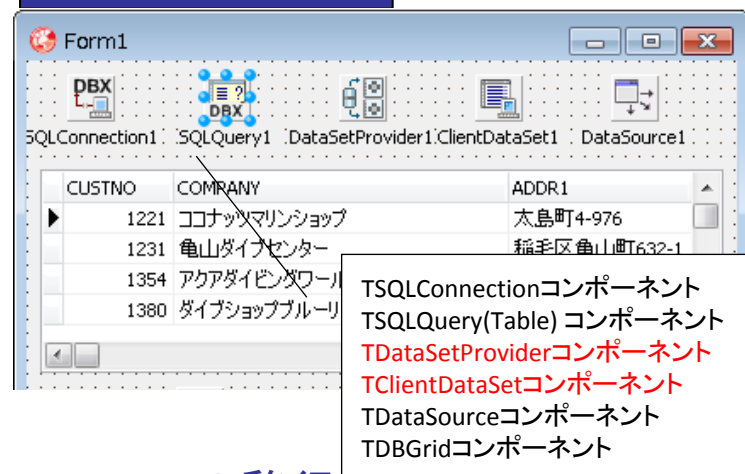
- BDEを使用している場合、変更するか継続するか。

## BDE から dbExpressに変更する一例

### BDE



### dbExpress



## ■ Borland Database EngineアプリケーションのdbExpressへの移行

[http://www.borland.com/resources/jp/pdf/products/delphi/delphi\\_bde\\_to\\_dbexpress\\_jp.pdf](http://www.borland.com/resources/jp/pdf/products/delphi/delphi_bde_to_dbexpress_jp.pdf)

# バージョンアップの移行ポイント

バージョンアップ時によくある変更点  
(よくある問合せ)

# バージョンアップの移行ポイント

## バージョンアップ時によくある変更点(～Delphi5からVerUp)

- ユニット再編

Delphi6の段階でCLX対応のために、ソースコードの再編が行われました。このポイントをまたぐバージョンアップを行う場合、uses節の修正がよく必要になります。

### uses節の初期状態(Delphi5)

**uses**

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;

### uses節の初期状態(Delphi7)

**uses**

Windows, Messages, SysUtils, **Variants**, Classes, Graphics, Controls, Forms, Dialogs;

新しいユニットの追加



# バージョンアップの移行ポイント

## バージョンアップ時によくある変更点(～Delphi5からVerUp)

- 設計用コードの分離

Delphi6からコンポーネントのソースコードで設計時と実行時のコードを明確に分離しなければならなくなりました。同時にデザインインターフェースユニットのユニット名も変更されました。

### 対象

uses節に**DsgnIntf**を含むDelphi5以前のコンポーネントソースコード

### 対処法

uses節の**DsgnIntf**を**DesignIntf**へ変更し、**DesignEditors**, **VCLEditors**, **RTLConsts**を追加し、パッケージのrequiresリストに**designide**を追加

# バージョンアップの移行ポイント

バージョンアップ時によくある変更点(～Delphi6からVerUp)

- **SQLClientDataSetコンポーネント**

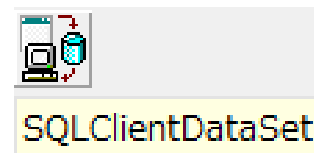
Delphi6にはdbExpressのコンポーネントでSQLClientDataSetがありますが、これはDelphi7以降では推奨されておらず、デフォルト組み込まれていません。

## 対象

SQLClientDataSetを使用しているプログラム

## 対処法

SQLClientDataSetは次のコンポーネントに分けて置き換え  
SqlQuery + DataSetProvider + ClientDataSet  
(SQLClientDataSetは元々これらのコンポーネントをセットにしたようなコンポーネント。SimpleDataSetに近い)



ソースはDemos関連のフォルダに互換用に格納されています。

..¥Demos¥Db¥SQLClientDataset ※バージョンによって若干異なります

バージョンによってコンパイルするために自身でソース修正が必要(非推奨)

# バージョンアップの移行ポイント

バージョンアップ時によくある変更点(～Delphi7からVerUp)

- **フォームの初期表示**

統合開発環境内でのフォームの表示方法が変更され、フォームの初期表示される位置が変更されました。

Delphi7以前のPositionプロパティ

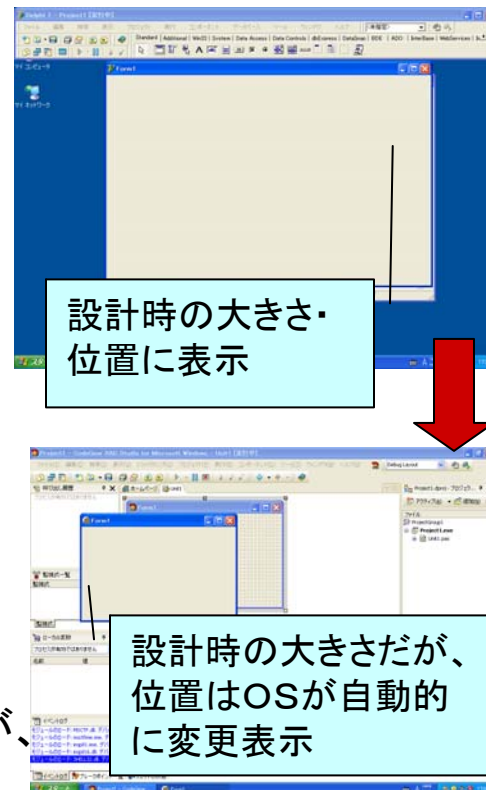
【poDesigned】

フォームは設計時と同じ画面上の位置、高さ、幅で表示

Delphi2005以降のPositionプロパティ

【poDefaultPosOnly】

フォームは設計時にフォームを作成したサイズで表示されるが、画面上の位置はオペレーティングシステムが選択



Positionプロパティのデフォルト値のままではフォーム情報(dfm)に保存されません。そのため、Delphi7以前でデフォルト値で作成したプログラムをDelphi2005以降で扱うと、デフォルト値が変わってしまうので注意が必要です。

# バージョンアップの移行ポイント

バージョンアップ時によくある変更点(～Delphi7からVerUp)

- **Fontプロパティのデフォルト値**

Fontプロパティのデフォルト値が変更されたため、  
フォントを指定していない場合は見た目が異なります。

## Delphi7以前のFontプロパティ

フォント名: **MS P ゴシック**      フォントサイズ: **9**  
CharSet: **SHIFTJIS\_CHARSET**

## Delphi2005以降のFontプロパティ

フォント名: **Tahoma**      フォントサイズ: **8**  
CharSet: **DEFAULT\_CHARSET**

MS P ゴシック、9ポイント

abcABCあいう

Tahoma、8ポイント

abcABCあいう



Delphi7以前で作成したプロパティ値が設計画面で  
引き継がれたコンポーネントはフォーム情報(dfm)に値が  
保存されているので問題ありません。ソース上で動的に作成している  
コンポーネントは明示的な指定などをしていない場合、注意が必要です。

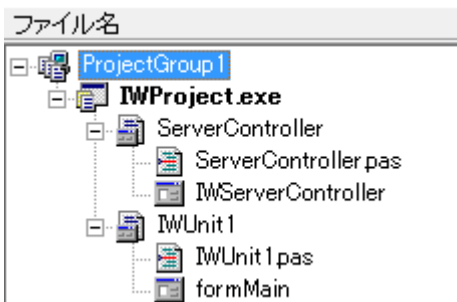
# バージョンアップの移行ポイント

## バージョンアップ時によくある変更点(～Delphi7からVerUp)

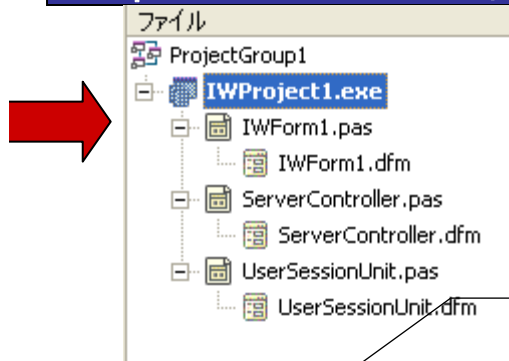
- **IntraWebプロジェクトは再作成を推奨**

Delphi7で作成したIntraWebのプログラムは、Delphi2005以降に付属するIntraWebのバージョンとはかなり大きな差異があります。構成やロジックがかなり違うため修正するよりも新規に再作成がお勧め。

### Delphi7付属IntraWebの基本構成



### Delphi2005以降の付属IntraWeb(VCL for the WEB)の基本構成



プロジェクトのファイル構成自体も異なるので修正してバージョンアップするのは難しい。

旧プロジェクトをバージョンアップでコンパイルすると、根本的に必要なユニットの不足や構造の違いが原因でエラーとなります。

エラー

- ① ユニット名 'IWInitStandAlone' を解決できません 行 5 (5:3)
- ② 未定義の識別子: 'IWRun' 行 12 (12:3)

メッセージ

[DCC エラー] IWProject.dpr(7): F1026 ファイル 'IWInitStandAlone.dcu' が見つかりません



# バージョンアップの移行ポイント

バージョンアップ時によくある変更点(～Delphi2006からVerUp)

- **Transparentプロパティのデフォルト値**

Transparentプロパティのデフォルト値が変更されたため、デフォルト値を使用している場合、意図せず透過表示になります。

Delphi2006までのTransparentプロパティ

False

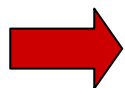
Delphi2007からのTransparentプロパティ

True

例: Labelコンポーネント

Delphi2006まではデフォルト値がFalse

得意先コード



Delphi2007からはデフォルト値がTrue

得意先コード

Falseに設定すれば  
本来の表示になります

Transparentプロパティのデフォルト値のままでは  
フォーム情報(dfm)に保存されません。

そのため、Delphi2006以前でデフォルト値で作成したプログラムを  
Delphi2007以降で扱うと、デフォルト値が変わってしまうので注意が必要です。



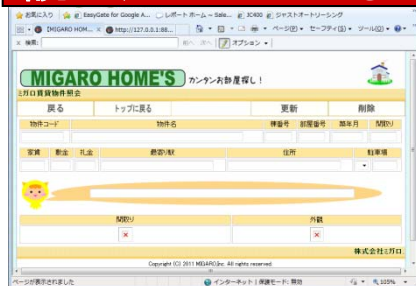
# バージョンアップの移行ポイント

## バージョンアップ時によくある変更点(～Delphi2007からVerUp)

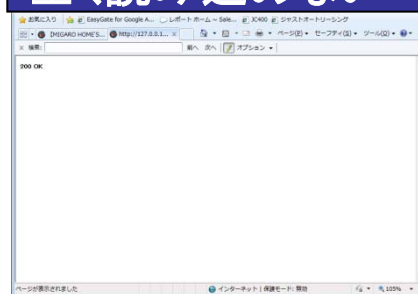
- **IWTemplateProcessorHTML**

IntraWeb(VCL for the WEB)でIWTemplateProcessorHTMLを使用してhtmlテンプレートを読み込んでいる場合、htmlの保存形式やhtml内のCharSetの設定をUTF-8に変更しておく必要があります。

**Delphi2007まで  
読み込めていても**



**Delphi2009～では  
全く読み込めない**



**対処法(読み込むhtml)**

**htmlファイル自体を  
文字コードUTF-8で保存**

文字コード(E): UTF-8 保存(S) キャンセル

3C//DTD HTML 4.01 Transitional//EN">

```
<!-- IBM WebSphere Studio Homepage Builder -->
<html content="text/html; charset=UTF-8">
<style-type="text/css">
<title>
貨物件照会</TITLE>
<link href="http://localhost/css/jacitr.css" type="te
```

**htmlソース内のCharSet  
をUTF-8に変更**

# バージョンアップの移行ポイント

## バージョンアップ関連情報

### バージョン移行/新機能/文字コード関連で役立つ情報URL

#### ■ Delphi 7以降の IDEの新機能

<http://edn.embarcadero.com/jp/article/34361>

#### ■ Delphi 2009 では始めるUnicode アプリケーション

<http://conferences.embarcadero.com/jp/article/images/39112/a6.pdf>

#### ■ Delphi 7ユーザと初心者のためのDelphi 2010入門

<http://edn.embarcadero.com/article/images/40483/1C.pdf>

#### ■ Delphi での文字コードのハンドリングについて

<http://edn.embarcadero.com/article/images/40483/1E.pdf>

#### ■ 旧Delphiで作成されたアプリケーションを Delphi 2010 に移行するには

<http://edn.embarcadero.com/article/images/40857/a3.pdf>

#### ■ 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>



# 補足資料2

## 旧DelphiをWindows7 へインストールする

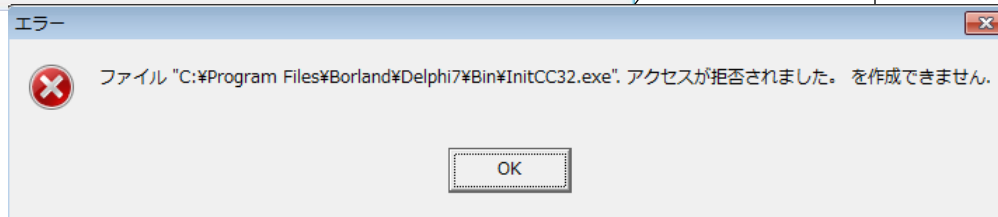


# 旧DelphiをWindows7へインストールする

- Windows 7へインストールする場合



エラー



素直にProgram FilesにインストールするとUACの関係で色々エラー原因になります。

# 旧DelphiをWindows7へインストールする

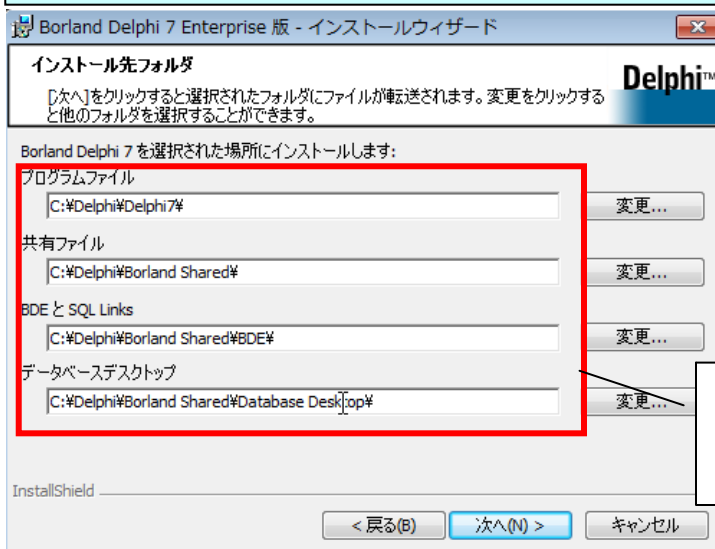
- Windows 7へインストールする場合

## 対処法

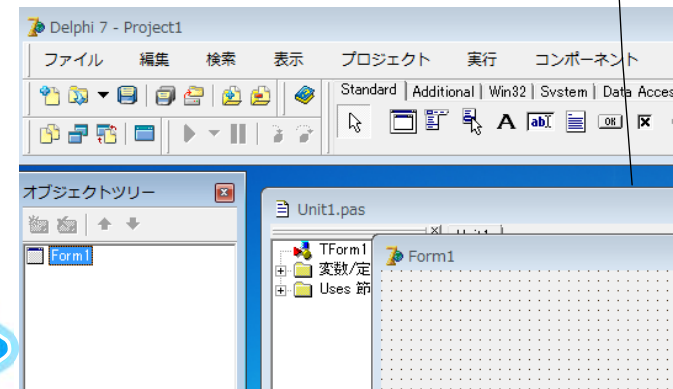
①インストール用のフォルダを作成する。  
例) C:\Delphi

C:\Delphi

②インストール時に①フォルダを指定する。  
UACに関わるフォルダは使わない。



例)  
Windows7でDelphi7を実行  
※正式な動作保障はありません。



インストールフォルダを  
指定

