### **embarcadero**<sup>.</sup>

17th Embarcadero Developer Camp

【B5】 Delphiチュートリアル セッション 「知って得する! 現役ヘルプデスクが答える Delphiテクニカルエッセンス 5.0」

#### 株式会社ミガロ. 吉原 泰介

⊘ (†) 🐼 ⊙ (J) 😕 🕶 🕸 (?)



# はじめに

⊘ (†) 🐼 😋 🕽 😕 😁 垫 🕐

アジェンダ

• ミガロ.について

よくある問合せQ&A

■Q1. VCL for the WEBでWebサービス連携

■Q2.トレイアイコンの利用

■Q3. OpenOfficeの利用



# ■XMLデータバインディングの使い方 ■ Delphi2009/2010 VCL for the WEB留意点



### ミガロ.について

7th Embarcadero Developer Camp

#### 株式会社ミガロ. -A 100% IBM i Company-



- Delphi/400
- DelphiをIBM i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



### **embarcadero**:

17th Embarcadero Developer Camp

### よくある問合せQ&A

!【質問】

VCL for the WEB (IntraWeb)で郵便番号検索サービスの 機能を組み込むことはできますか?



Webサービスを利用することで郵便番号検索の機能を そのまま利用することができます。 Webサービスの提供内容によって、Delphiからの利用の 方法も異なります。 XMLで結果を受け取れる形であれば、XMLデータ バインディングの機能で容易に実装することができます。



th Embarcadero

7th Embarcadero Developer Camp

#### Webサービス連携実装例(郵便番号検索)

			Ft X	🕒 Bing		
× Norton · 🕜 · 🕑 カードとロ!	イン情報・					
🚖 お気に入り 🍰						
🚳 取引先データ入力			🏠 🕶 🗟 👻 🖃	🖶 ▼ ページ(P) ▼	セーフティ( <u>S</u> ) ▼	ツール <b>(<u>0</u>) ▼</b>
MIG						
コード: 00	0001					
名称:株	式会社ミガロ.		①郵便	、 奋 亏 を 人	、フ	
<b>郵便番号</b> : 55	6					
5		在士阪古泊海区	口大场古		*	
54	60008、大阪) 60004、大阪!	有大阪市浪速区	口 中 日 本 陸 西			
	60017:大阪	府大阪市浪速区	海町			
55	60016 : 大阪)	府大阪市浪速区	元町		(≡)	
都道府県:		地図表	赤			
市区町村:						
町域等:						
	/	$\bigcirc \mathbb{N} / \mathbb{A} + \mathbb{H}_{-}$	ービフト			
保友 取	当			<i></i>	_	
		前方一致(	の住所り	ストを取得	导	
						l
ページが表示されました		8	😜 インターネット	保護モード: 無効		• 🔍 100%

#### 今回利用する郵便番号検索Webサービス

グルーブテクノロジー社提供Webサービス http://groovetechnology.co.jp/webservice/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。



Combarcadero \*

h Embarcadero

#### 郵便番号検索サービス連携の仕組み

郵便番号(TIWEdit)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。

h Embarcadero



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。





Embarcadero

#### 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。



h Embarcadero

#### 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

?以降がパラメータ部分 Format:返却形式 le: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。



h Embarcadero

#### 住所データ取得(ソース) procedure TfrmWebSample2Input.edtMCYUBNAsyncExit(Sender: TObject; EventParams: TStringList); var i: Integer; URI: String; RootNode: IXMLGroovewebserviceType; //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を処理するリクエストURL編集 XMLDocument1, XML, Text := IdHTTP1, Get(URI); RootNode := Getgroovewebservice(XMLDocument1); |stMCYUBN.|tems.Clear; // 住所リストをクリア ソース内コンポーネント名 // 受け取ったXMLデータから住所リストを作成 ■郵便番号入力 for i := 0 to RootNode. Address. Count - 1 do iedtMCYUBN : TIWEdit IstMCYUBN. Items. Add (Format ('%d : %s%s%s'. ■住所リスト表示 [RootNode. Address. Items[i]. Zipcode, //郵便番号 IstMCYUBN : TIWListbox RootNode. Address. Items[i]. Prefecture, //都道府県名 ■処理用 RootNode. Address. Items[i]. City, //市町村名 XMLDocument1 : TXMLDocument RootNode. Address. Items[i]. Town])); //町域名 IdHTTP1 : TIdHTTP // 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 IstMCYUBN. Items. Count > 0 then IstMCYUBN. ItemIndex := 0: end:

Embarcadero



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。





Embarcadero

7th Embarcadero Developer Camp



16

#### 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** <u>http://code.google.com/intl/ja/apis/chart/index.html</u> 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。

h Embarcadero



#### 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。

<u>h Embarcadero</u>

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



#### 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:<u>http://chart.apis.google.com/chart?</u>
  - ・サイズ:<u>chs=500x200</u>
  - •割合:<u>chd=t:60,40</u>
  - ・グラフの種類:<u>cht=p3</u>
  - ・ラベル: <u>chl=Hello</u>World



h Embarcadero

呼び出し構造は <u>共通アドレス?(パラメータ1)&(パラメータn)</u>・・・ 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



#### 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=IC



A http://chart.apis.google.com/chart?chs=30...

🏉 http://chart.apis.google.com/chart?chs=30...

http://chart.apis.google.com/chart?chs=300x300

<u>&chd=t:30,20,60,80</u>&**Cht=bvs** 



Embarcadero

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。

h Embarcadero

21



#### 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%	680%80%E5%86%85
■グラフの種類(棒グラフ)	■Xラベル
cht=bvg	chxl=0: 2001 2002 2003 ····
■サイズ	■Y目盛
chs=300x300	chxr=1,0,2000
■色	■最小/最大値
chco=76A4FB	chds=0,2000
■ラベル表示	■データ
chxt=x,y	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。



n Embarcadero





Embarcadero





Embarcadero

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.IWAppFormCreate(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;
```



h Embarcadero

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。

🎑 Fo	rm1														, [		<u> </u>
								M	•		•		H				
																	_
コート	゛地域	地区	2001年	2002年	2003年	2004年	2005年 🔺										*
01	北海道	稚 内	1088	1111	804	1107	10					根	室				
02	北海道	旭 川	1054	1273	988	1207	10	2000									
03	北海道	根 室	1357	1211	3000	917	10	1800-									
05	北海道	帯 広	1091	755	1022	938	10										
06	北海道	函館	1511	1225	1200	1000	13 🗏	1600-									
07	東北区	青 森	1290	1028	1095	1010	12	1400-									
08	東北区	秋田	1893	1594	1915	1231	19	4000									
09	東北区	盛岡	1702	1044	1206	828	14	1200-									
10	東北区	仙台	1337	977	1466	1239	ç	1000									
11	東北区	山 形	992	938	1198	901	12	000									
12	東北区	福島	1056	1066	1307	925	11	800									
13	関東区	宇都宮	1634	1373	1461	1154	14	600-									
14	関東区	前橋	1201	1174	1347	971	10	400-									
15	関東区	水戸	1491	1328	1394	1237	12	400									
16	関東区	熊 谷	1540	1440	1355	1048	11	200-									
17	関東区	銚 子	1611	1783	1787	1357	14	0									
18	関東区	東 京	1513	1620	1873	1131	12	, i i	2001	2002	2003	2004	2005	2006	2007	2008	
	·																

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。

arcadero

Embarcadero

Q2 トレイアイコンの利用



#### Windowsのトレイ通知やトレイアイコンへの常駐といった アプリケーション操作はできますか?



DelphiではVersion2006よりTTraylconという トレイ操作専用のコンポーネントが提供されています。

#### 💏 TTrayIcon

これを利用するとWindowsのトレイ通知やトレイへの 常駐といったアプリケーション動作が可能です。



Q2 トレイアイコンの利用

28

トレイアイコンとは デスクトップの右下のタスクトレイに格納された常駐している アプリケーションのアイコンです。 以下WindowsXPでのトレイアイコンの画面例

アプリケーションを終了させずに常駐させる使い方

ウィルス対策ソフトなどは タスクトレイに常駐

必要なメッセージを通知したり、

時間のかかるプログラムをバックグラウンドで処理する使い方

ウイルスバスター2010により保護されています

😯 🍯 😫 🕗 ‰ 🔊 🚳 17:43



Q2 トレイアイコンの利用

#### ×ボタンでアプリケーションを終了させずに常駐させる使い方例





#### フォームのCloseQueryイベントでアプリケーション終了させずに 常駐させる(ソース) procedure TForm1.FormCloseQuery(Sender: TObject; var CanClose: Boolean); begin //閉じるボタン以外でCloseされる場合はトレイアイコンへ ソース内コンポーネント名 if ActiveControl $\Leftrightarrow$ btnClose then Traylcon1 : TTraylcon begin CanClose := False; //画面を閉じない Self.Visible := False; //画面非表示 TrayIcon1. Visible := True; //トレイアイコン表示 TrayIcon1. BalloonHint := 'タスクトレイに常駐しています': //メッセージ設定 Travicon1. ShowBalloonHint; //メッセージ表示

トレイアイコンをクリックされたら元の画面を表示(ソース)

procedure TForm1.Traylcon1Click(Sender: TObject);	サンプルコードでは
begin Self.Visible := True; //画面表示 Traylcon1.Visible := False; //トレイアイコン非表示 end;	Traylconを直接操作してい ますが、一連の処理を関数 化しておけば便利です。

## Q2 トレイアイコンの利用

end:

end;

#### 'th Embarcadero evelope<u>r Camp</u>

タスクトレイに常駐しています

🚮 🕨 🛱 🔥 ...l

Q2 トレイアイコンの利用

#### 時間のかかる処理をバックグラウンド化する使い方例









時間のかかる処理でトレイアイコンに常駐する(ソース)

Q2 トレイアイコンの利用

Q2 トレイアイコンの利用

常駐監視しながらメッセージ通知する使い方例



## Q2 トレイアイコンの利用

#### 7th Embarcadero Jeveloper Camp



procedure TForm1.FormCloseQuery(Sender: TObject; var CanClo	se: Boolean);
begin	
//閉じるボタン以外でCloseされる場合はトレイアイコンへ	いーフロコンポーネントタ
if ActiveControl <> btnClose then	
begin	IIImeri : IIImer
CanClose := False; //画面を閉じない	
Self.Visible := False; //画面非表示	
Traylcon1.Visible := True; //トレイアイコン表	示
Traylcon1. BalloonHint := 'タスクトレイに常駐しています	'; //メッセージ設定
Traylcon1.ShowBalloonHint; //メッセージ表示	
Timer1.Enabled := True; //タイマー監視開始	タイマー制御
end;	
end;	

#### トレイアイコンをクリックされたら元の画面を表示(ソース追記)







#### TTimerで通知対象データの監視(ソース)

Q2 トレイアイコンの利用

17th Embarcadero Developer <u>Camp</u>

7th Embarcadero Developer Camp



DelphiからOpenOfficeをExcelのように利用することは できますか?



OpenOfficeのCalcもOLEで操作することができるので、 Excelと同様にセル編集やファイル出力といった動作を実 装することができます。 またExcelにはないPDF出力などの便利な機能も利用す ることができます。 ただしxls形式ファイルを読み込み/出力も可能ですが、 Calcの機能上、xlsx形式での出力は行えません。



7th Embarcadero Developer Camp

OpenOffice(正式名称: OpenOffice.org)とは オープンソースのオフィスソフトの一つ。 Microsoft社の「Microsoft Office」と高い互換性があり、 LinuxやWindowsなど様々なOSで動作する。 オープンソースソフトのため、誰でも自由に無償で利用することが でき、また、興味があれば開発に参加することもできる。 (IT用語辞典引用)

ja: OpenOffice.org 日本語プロジェクト http://ja.openoffice.org/



7th Embarcadero Developer Camp

OpenOfficeで使用できる主な機能

Writer ワープロソフト 数式作成ソフト Math 図形描画ソフト 表計算ソフト Calc Draw Impress プレゼンテーションソフト Base データベースソフト 舞舞 1 - OpenOffice.org Cale ファイル(F) 編集(E) 表示(V) 挿入(I) 書式(O) ツール(T) データ(D) ウィンドウ(W) 今回は表計算のCalcを利用 見た目も操作もほぼMicrosoftのExcelと同じです。 ファイル拡張子は.odsです。xlsやxlsxも読み込めます。 無償で使うことができますが、もちろんオープンソース なので正式な製品サポートがあるわけではありません。 標準 100%

7th Embarcadero Developer Camp

#### 以降のDelphiからのCalc操作のサンプルは OpenOffice.org 3.2.1をインストールした環境での動作です。

🏨 OpenOffice_C 👝 💿 💌	m 無題 2 - OpenOffice.org(	alc			
基本操作	ファイル(E) 編集(E) 表示	(*) 挿入(1) 書式(0) ツー	·ル(T) データ(D) ウィンドウ(M	) ヘルプ(H)	2
			▌▌▘▓▕▋▘▓▕▌▖ख़▝▕▇	2• x• 100 27 100 143 114 1 1,6 % %	
起動	A1 •	x ∑ = [			
終了		C D	E F	G H	🔁 aaaa.pdf - Adobe Reader
1-2-1	2	B5 Delph:	セッション		ファイル(E) 編集(E) 表示(V) 文書(D) ツール(I) ウィンドウ(W) ヘルプ(H)
編集操作	4 5 6				Image: Image
	7 8				— 读 検索 ▼
セルの書込み	10				
フォント変更	12 13				■ ■ D展デベロッパーキャンプ
	15				B5 Delphiセッション
事線を引く	17 18				E
	20 21				
出力操作	ぼ     )     )     )     )       ジート1/3     )	ーキャンプ (Sheet2) ( 黒準 )	···· 栗準 (*) 合計=0	<b>⊖</b> — •	
保存					
Excel保存					<i>D</i>
PDF保存					
印刷					•
					(mbarcadoro' 39

17th Embarcadero Developer Camp

今回のサンプルでは処理をボタンで分割しているので サービス、ファイル関係の変数はグローバルで定義します。

#### Calc操作を行うための共通変数・関数(ソース)

private vOpenOffice : Variant; //サービスマネージャ vStarDesktop: Variant; //サービス vDocument : Variant; //ドキュメント :Variant; //シート vSheet function dummyArray: Variant; //空プロパティ配列設定用 //空のプロパティ配列を返却 function TForm1. dummyArray: Variant; begin Result:= VarArravCreate([0, -1], varVariant); end:



#### 7th Embarcadero Developer Camp

41

#### CalcのAPIでのPropertyValue

 Calcの操作上で使用するAPIのパラメータにはPropertyValueという 構造体がよく使われます。
 このパラメータは、ドキュメントのオープンと作成に対するさまざまなオプションを OpenOffice.org に提供します。



ファイル保存のAPIを利用する例
//パラメータ用プロパティ配列生成
vaProperties := VarArrayCreate([0, 0], varVariant);
//プロパティ用オプション生成
vProperty:= vOpenOffice.Bridge\_GetStruct('com.sun.star.beans.PropertyValue');
//プロパティ用オプション値設定
vProperty.Name := 'FilterName';
vProperty.Value := 'MS Excel 97';
//オプション内容をプロパティ配列に設定
vaProperties[0] := vProperty;
//ファイルを保存
vDocument.StoreToURL('file:///' + sFileName, vaProperties);

17th Embarcadero Developer Camp

CalcのAPIでのPropertyValue(続き)

 ただPropertyValue構造体がパラメータの場合でも、 オプションを特に設定する必要がないのであれば 毎回Var配列を定義するのも面倒なので、先に実装した dummyArrayで代用します。





#### 17th Embarcadero Developer Camp

#### Calcを終了する(ソース)





Calcでファイルを保存する(ソース)



### Calcでxls保存する(ソース)

```
procedure TForm1. btnCalcXlsClick(Sender: TObject);
var
 vaProperties : variant; //保存パラメータ用<配列>
 vProperty : variant; //保存オプション用<配列値>
 sFileName : String; //保存ファイル名編集用
begin
 //保存ダイアログ実行
 if SaveDialog1. Execute then
 begin
   //ダイアログの保存パスを/形式に置換
   sFileName := StringReplace(SaveDialog1.FileName, '¥', '/', [rfReplaceAll]);
   //パラメータ用プロパティ配列生成
   vaProperties := VarArrayCreate([0, 0], varVariant);
   //プロパティ用オプション生成
   vProperty:= vOpenOffice.Bridge GetStruct('com.sun.star.beans.PropertyValue');
   //プロパティ用オプション値設定
                                                     基本は

ods

と同じ
   vProperty.Name := 'FilterName';
   vProperty. Value := 'MS Excel 97';
                                                     ですが、xlsで保存する
   //オプション内容をプロパティ配列に設定
                                                     場合は別形式なので
   vaProperties[0] := vProperty;
                                                     フィルタープロパティで
   //ファイルを保存
                                                     指定する必要があります。
   vDocument. StoreToURL ('file:///' + sFileName, vaProperties);
                                                    XIsx形式保存はCalcでは不可
   //変更保存等ダイアログを表示しない
   vDocument Modified := False:
                                                                            47
 end:
                                             保存APIはStoreToURLで
end:
```

#### 🔁 aaaa.pdf - Adobe Reader ファイル(E) 編集(E) 表示(Y) 文書(D) ツール(I) ウィンドウ(W) CalcでPDF保存する(ソース) ヘルプ(日) 🚔 🔬 र 🌄 🔶 🕂 / 1 💌 🖲 40.5% र procedure TForm1. btnCalcPDFClick(Sender: TObject); 😝 🔂 検索 var vaProperties : variant; //保存パラメータ用<配列> B5 Delphiセッション vProperty : variant; //保存オプション用<配列値> sFileName : String; //保存ファイル名編集用 begin //保存ダイアログ実行 if SaveDialog1. Execute then begin //ダイアログの保存パスを/形式に置換 sFileName := StringReplace(SaveDialog1.FileName, '¥', '/', [rfReplaceAll]); //パラメータ用プロパティ配列生成 vaProperties := VarArrayCreate([0, 0], varVariant); //プロパティ用オプション生成 vProperty:= vOpenOffice.Bridge GetStruct('com.sun.star.beans.PropertyValue'); //プロパティ用オプション値設定 基本は ods と 同じ vProperty.Name := 'FilterName'; ですが、pdfで保存する vProperty. Value := 'calc pdf Export'; //オプション内容をプロパティ配列に設定 場合は別形式なので vaProperties[0] := vProperty; フィルタープロパティで //ファイルを保存 指定する必要があります vDocument.StoreToURL('file:///' + sFileName, vaProperties); 48 end; 保存APIは必ずStoreToURLで end:



7th Embarcadero Developer Camp

mbarcadero

50



#### 7th Embarcadero Developer Camp

#### Calc上での操作のコーディングが分からないときは・・

ツール(T) データ(D) ウィンドウ(W) へ	ルプ(出)		REM **** BASIC ****
🍪 スペルチェック(S) F7 🛃	i 🔟 🥢 👬 🧭 💼	🗟 Q, 🕜	
言語(B) , (二)	▲		sub Main
トレース(1) , 🗐	گٽ ڏيٽ <sup>2</sup> ي کھ اولي ا		rem ————————————————————————————————————
🌴 ゴールシーク(G)			dim document as object
	; Н	I J K	dim dispatcher as object
シナリオ(E)			rem get access to the document
ドキュメントの共有(H)			document = ThisComponent.CurrentController.Frame
ドキュメントの結合(T)			dispatcher = createUnoService(
ドキュメントの保護(P)			rem
			dim args1(0) as new com.sun.star.beans.PropertyValue
			args (0) Name - Scringhame args 1(0) Value = "あいうえお"
💼 ギャラリー(G)			
メディアプレイヤー(Y)			dispatcher.executeDispatch(document, .uno:EnterString , , 0, args1())
	マクロの記録(A)		rem
拡張機能マネージャー(E)	マクロを実行 <b>(U)</b>		dispatcher.executeDispatch(document, ".uno:JumpToNextCell", "", 0, Array())
<u>X</u> ML フィルタの設定	マクロの管理(0)	OpenOffice.org Basic(B).	
L		1	end sub
		¬ /	⑥操作のマクロが
「ラフクロの答	田から	/	
	埋から		コーナインクの参
保存したマク	ロを開く		キにできます
			方にじさまり

### **embarcadero**<sup>.</sup>

17th Embarcadero Developer Camp



# 補足資料 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使い方

XMLデータバインディングの利用① (以前もやった覚えが・・

 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

🥪 Delphi 2010	
ファイル 編集 検索 表示 リ	ファクタリング プロジェクト 実行 コンパ
<ul> <li>新規作成</li> <li>▶</li> <li>□</li> <li>□&lt;</li></ul>	<ul> <li>         ・ パッケージ - Delphi         <ul> <li>ユニット - Delphi             </li> <li></li></ul></li></ul>
	□ フォーム - Delphi □ フォーム - Delphi □ フォーム - Delphi □ Delphi プロジェクト □ Q 検索
<ul> <li>■ 「「古き休存 Ctrl+S</li> <li>③ 名前を付けて保存…</li> <li>■ プロミッテクト(一々前をけ(+) ア)四ち</li> </ul>	その他… カスタマイズ… DataSnap Server Delphi ファイル VCL for the Web WebBroker WebSnap

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使い方

#### 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ウ・	ィザー	- ř		¢	XML データバインディングウィザード	• المحالي محالي محال
スキーマコンボーネント( <u>C</u> ):	• III	ソース名( <u>5</u> ): ソース データ型( <u>1</u> ) ドキュメントを作成( <u>D</u> )	groovewebserviceType ComplexType マバインディングを生成(G)		バインドの一覧(5) 生成されたインターフェイス ② IXMLGroovewebserviceType ② IXMLAddressType ③ IXMLOfficeType	生成コードのプレビュー(C): { IXMLGroovewebserviceType } IXMLGroovewebserviceType = interface(IXMLNode) ['{7D5F9AC2-974C-4B1F-9E7A-2ACA1B929358}'] { プロパティ参照関数 } function Get_Patetime: UnicodeString; function Get_Datetime: UnicodeString; function Get_Address: IXMLAddressTypeList; function Get_Office: IXMLOfficeTypeList; procedure Set_Version(Value: UnicodeString);
<ul> <li>town</li> <li>prefecture_yomi</li> <li>city_yomi</li> <li>town_yomi</li> <li>officeType</li> <li>zipcode</li> <li>prefecture</li> </ul>	Ŧ	識別子名( <u>1</u> ): 要素名( <u>L</u> ):	IXMLGroovewebserviceType 「ドキュメント要素型(E) groovewebservice		<ul> <li>データバインディングの設定(5)</li> <li>設定を保存しな(、v(M)</li> <li>スキーマファイルに保存(X)</li> <li>ファイルに保存(I):</li> <li>C:¥Temp¥zipsearch.xdb</li> </ul>	<pre>Flocted and Set_Datetine (and et childode string); {メソッドとプロパティ} property Version: UnicodeString read Get_Version write Set_' property Datetime: UnicodeString read Get_Datetime write Si property Address: IXMLAddressTypeList read Get_Address; property Office: IXMLOfficeTypeList read Get_Office; end;</pre>
7	プション	/( <u>(</u> ) < 戻る(f)	次へ(11) > キャンセル			)) < 戻る(P) 完了(F) キャンセル ヘルブ(H)



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。

<i></i>	**************************************
【 【 / XMLデー	-タバインディング ) 
(作成日:2009/12 作成元:Webサー (設定ファイルの保管先:同一フォ (****************************	/01 00:00:00 '} ビスから得られるXMLデータ   + ルダ内 (zipsearch.xdb)     **********************************
aunit zipsearch;	
interface uses xmldom, XMLDoc, XMLIntf;	生成されたソース
type (前方宣言)	
IXMLGroovewebserviceType = inter IXMLAddressType = interface; IXMLAddressTypeList = interface; IXMLOfficeType = interface; IXMLOfficeTypeList = interface;	
{ IXMLGroovewebserviceType }	
IXMLGroovewebserviceType = inter ['{C156869D-BC0C-475B-A5F6-BD3 {プロバティ参照関数} function Get_Version: WideStri function Get_Datetime: WideStri function Get_Address: IXMLAddri function Get_Office: IXMLOffic procedure Set_Version(Value: W procedure Set_Datetime(Value: W property Version: WideString)	face(IXMLNode) C6606A230}'] ing; essTypeList; eTypeList; ideString); WideString); ead Get_Version <b>trite</b> Set_Version; read Get_Natatime <b>trite</b> Set_Datatime.

Q Image: SampleChart.pas (C:¥サンプルプログラム¥DevCamp201009¥Q1¥WebSample2) Image: SampleFrm2.pas (C:¥サンプルプログラム¥DevCamp201009¥Q1¥WebSample2) Image: SampleFrm2.pas (C:¥サンプルプログラム YDevCamp201009¥Q1¥WebSample2) YDevCamp201009¥Q1¥WebSample2)
i SampleChart.pas (C:¥サンプルプログラム¥DevCamp201009¥Q1¥WebSample2) SampleFrm2.pas (C:¥サンプルプログラム¥DevCamp201009¥Q1¥WebSample2) zipsearch.pas (C:¥サンプルプログラ) ¥DevCamp201009¥Q1¥WebSample2)
追加先:
OK         キャンセル

### **embarcadero**:

17th Embarcadero Developer Camp





## Delphi2009/2010 VCL for the WEB留意点

9 (h) 🐼 😁 J) 😕 😁 🏝 📍

実行表示時の0Dを含む文字の欠落

Unicodeの下2桁が0Dの文字("名","不", "損","服","納","鈍", "倍"・・・等) はTIWLabelのCaptionプロパティやTIWEditのTextプロパティで設定すると実 行時に表示が欠落してしまいます。



これは0Dが改行コードとして認識されているために発生しているので 回避対応が必要です。

#### QC報告 http://qc.embarcadero.com/wc/qcmain.aspx?d=87756



Embarcadero

実行表示時の0Dを含む文字の欠落の対処方法例

 TIWLableコンポーネント RawTextプロパティをTrueに設定すると改行の制御がOffになるので 正しく表示できます。

納品先名

Embarcadero

品先	

 TIWEditコンポーネント RawTextプロパティはないので、IWCompEdit.pasに対応を加えて ユニットを組み込めば正しく表示できます。

IWCompEdit.pasの対応点 TIWCustomEdit.RenderHTML関数内の次の部分を変更 Result.AddStringParam('VALUE', TextToHTML(Text)); → Result.AddStringParam('VALUE', TextToHTML(Text, False, False));



Asyncイベント実行時の0Dを含む文字の欠落

Async関係のイベントで0Dの文字をTIWEditのTextプロパティに設定する と表示が欠落してしまいます。



これはRenderHTML関数ではなくRenderAsync関数で欠落しているため 前述の対応は有効ではありません。



Embarcadero

Asyncイベント実行時の0Dを含む文字の欠落の対処方法例

TIWEditコンポーネント
 Async以外のイベントで処理を代行するか、
 IWCompEdit.pasに対応を加えてユニットを組み込む必要があります。

Embarcadero

IWCompEdit.pasに対応を加える場合は TIWCustomEdit.RenderAsync関数が修正対象になります。

■修正例■ 画面上に表示されているElement値は欠落していますが、 コンポーネント内で text として正しい値を保持しているので、 JavaScritpを発行して直接Element値を上書きします。



end;

end;

```
Asyncイベント実行時の0Dを含む文字の欠落の対処方法例
```

Embarcadero

```
function TIWCustomEdit.RenderAsync(AContext: TIWBaseHTMLComponentContext): TIWXMLTag;
var
 iwedtOwner: TComponent; //Ajax対応処理 add
                          //Ajax対応処理 add
 sText.sJS
           : String;
begin

    ・・・・既存ステップは掲載割愛・・・・

 RenderAsyncCommonProperties( AContext, Result );
 //Ajax対応処理 start
                                                  JavaScript発行のため、IWEdit
  iwedtOwner := Owner;
                                                  が配置されているIWFormを
  if Owner is TFrame then
                                                  Ownerとして取得。
  begin
                                                  Frameが使われた場合を考慮し
   while (iwedtOwner is TFrame) do
                                                  てループ処理。
   begin
    iwedtOwner := TFrame(iwedtOwner).Owner;
```

#### Asyncイベント実行時の0Dを含む文字の欠落の対処方法例

arcadero

