

【セッションNo. 2】

Delphi/400 最新技術情報

Delphi/400で本格Webアプリ開発

株式会社ミガロ.

RAD事業部 技術支援課

吉原 泰介

## 【アジェンダ】

- Delphi/400で開発するWebアプリケーション
- ここまでやりたい！本格Webアプリケーション開発
  - 機能にこだわる本格Webアプリケーション！  
動的明細、ポップアップ画面、PDF出力の実装
  - 画面にこだわる本格Webアプリケーション！  
html連携、jQueryMobileの実装
- まとめ

### ■お持ち帰り資料

- VCL for the Web基礎知識
- Webアプリケーションでの PDF出力方法

## ■ Delphi/400で開発するWebアプリケーション

### • Webブラウザで稼動するアプリケーションの魅力

インターネット環境とブラウザがあれば、利用可能。

→ クライアントにアプリケーションの配布が不要。

→ 社外や取引先でもインターネットを使って利用できる。

→ PCのOSや端末を問わずに利用できる。



Webアプリケーション

社外から  
インターネット  
で利用できる

運用管理の  
手間が少ない

環境ごとにアプリ  
ケーションを開発  
しなくてもOK

Windows以外  
のクライアント  
でも利用

スマートフォンでも  
タブレットでも利用

iPhoneでも  
Androidで  
も利用

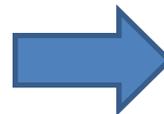
Delphi/400では『VCL for the Web』機能で  
Webアプリケーションを開発できる！

# ■ Delphi/400で開発するWebアプリケーション

## • VCL for the Web (IntraWeb)

Delphi/400に用意されたC/Sアプリと同手法でWEBアプリが開発できる機能

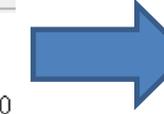
### ①プロジェクトの作成



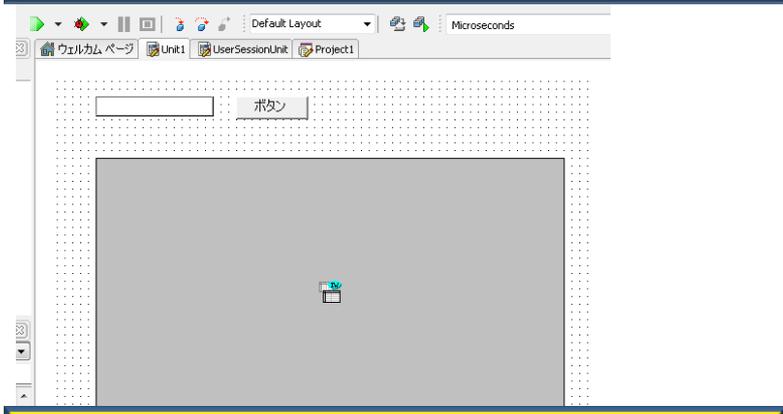
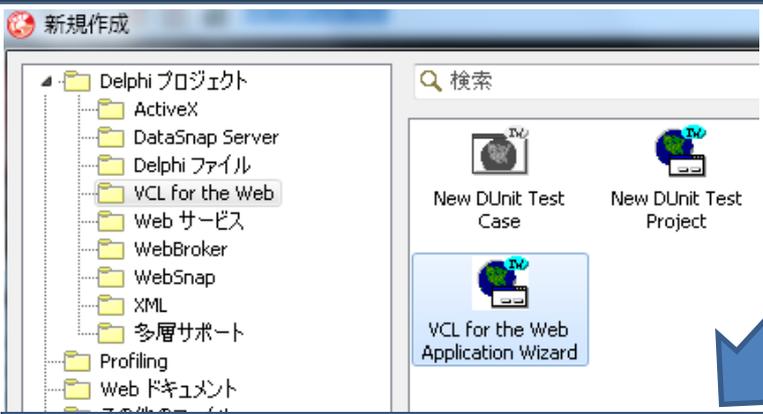
### ②コンポーネントの配置



### ③プログラムの記述



### ④コンパイル&実行



```
procedure TIWForm1.IWButton1Click(Sender: TO
begin
  WebApplication.ShowMessage(IWEdit1.Text);
end;

initialization
  TIWForm1.SetAsMainForm;

end.
```



# ■ Delphi/400で開発するWebアプリケーション

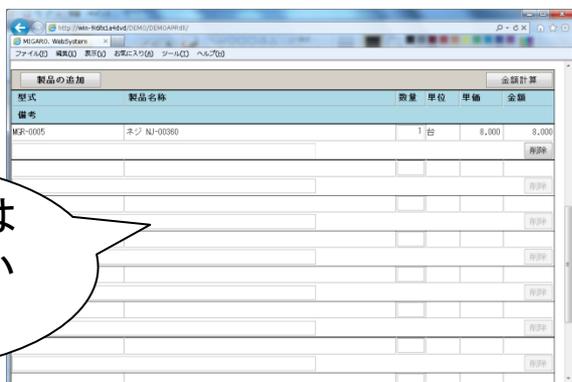
## • Webアプリケーションで実現したい機能

Webサイトで見たことがある便利な画面、機能・・・

Delphi/400のWeb開発でどうやって実現するのか分からない。

そんな疑問にお応えする本格Web開発テクニックをご紹介します！

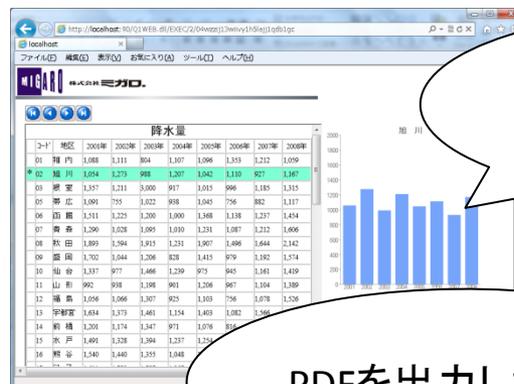
明細データは  
固定行でしか  
作れない？



サブ検索画面  
をポップアップ  
で使いたい！



もっとデザインに  
こだわりたい！



PDFを出力したい！

日	売上	粗利	売上高	粗利益	売上高	粗利益
01	1,200	300	1,200	300	1,200	300
02	1,500	375	1,500	375	1,500	375
03	1,800	450	1,800	450	1,800	450
04	2,100	525	2,100	525	2,100	525
05	2,400	600	2,400	600	2,400	600
06	2,700	675	2,700	675	2,700	675
07	3,000	750	3,000	750	3,000	750
08	3,300	825	3,300	825	3,300	825
09	3,600	900	3,600	900	3,600	900
10	3,900	975	3,900	975	3,900	975
11	4,200	1,050	4,200	1,050	4,200	1,050
12	4,500	1,125	4,500	1,125	4,500	1,125
13	4,800	1,200	4,800	1,200	4,800	1,200
14	5,100	1,275	5,100	1,275	5,100	1,275
15	5,400	1,350	5,400	1,350	5,400	1,350
16	5,700	1,425	5,700	1,425	5,700	1,425
17	6,000	1,500	6,000	1,500	6,000	1,500
18	6,300	1,575	6,300	1,575	6,300	1,575
19	6,600	1,650	6,600	1,650	6,600	1,650
20	6,900	1,725	6,900	1,725	6,900	1,725
21	7,200	1,800	7,200	1,800	7,200	1,800
22	7,500	1,875	7,500	1,875	7,500	1,875
23	7,800	1,950	7,800	1,950	7,800	1,950
24	8,100	2,025	8,100	2,025	8,100	2,025
25	8,400	2,100	8,400	2,100	8,400	2,100
26	8,700	2,175	8,700	2,175	8,700	2,175
27	9,000	2,250	9,000	2,250	9,000	2,250
28	9,300	2,325	9,300	2,325	9,300	2,325
29	9,600	2,400	9,600	2,400	9,600	2,400
30	9,900	2,475	9,900	2,475	9,900	2,475
31	10,200	2,550	10,200	2,550	10,200	2,550
合計	150,000	37,500	150,000	37,500	150,000	37,500

スマートフォンの  
Web画面ってどう  
やって作るの？



# ■ ここまでやりたい！本格Webアプリケーション開発

## 機能にこだわる本格Webアプリケーション！

1.動的な明細画面の実装

2.ポップアップ画面の実装

3. PDF出力機能の実装

お持ち帰り資料

## 画面にこだわる本格Webアプリケーション！

4.htmlを連携したWeb画面実装

5.jQueryMobileを利用したスマートデバイスWeb画面実装

# ■ ここまでやりたい！本格Webアプリケーション開発

## 機能にこだわる本格Webアプリケーション！

### 1.動的な明細画面の実装

# ■ ここまでやりたい！本格Webアプリケーション開発

## 1.動的な明細画面の実装

Webアプリケーションの明細行入力画面ではあらかじめ用意した固定行数の明細入力で実装することが多い。

MIGARO. WebSystem

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

製品の追加

型式	製品名称				
MGR-0005	ネジ NJ-00360	1	台	8,000	8,000
					削除

あらかじめ用意した固定行数の明細入力や画面遷移を挟んでの明細入力

# ■ ここまでやりたい！ 本格Webアプリケーション開発

## 1. 動的な明細画面の実装

VCL for the Webでフレームを使えばC/Sアプリケーション同様画面上での自由な明細制御が実現できる！

The screenshot shows a web browser window displaying the 'MIGARO. Technical Seminar' website. The page is titled '注文入力画面' (Order Entry Screen). The header includes the MIGARO logo, the text 'MIGARO. Technical Seminar', and a login status 'ログイン中 株式会社ミガロ様' with a 'ログアウト' button. The main content area contains a form for entering order details, including fields for '会社名' (Company Name), 'ご担当者' (Responsible Person), '納入先郵便番号' (Delivery Postcode), '納入先住所' (Delivery Address), '納入先電話番号' (Delivery Phone Number), and '希望納品日' (Desired Delivery Date). A '合計金額' (Total Amount) of '¥1,050' is displayed. Below the form is a table titled '注文明細' (Order Details) with columns for '商品コード' (Product Code), '商品名' (Product Name), '数量' (Quantity), '単価' (Unit Price), and '金額' (Amount). The table contains two rows, each with a search icon and a '削除' (Delete) button. A red box highlights the table area, and a callout box points to it with the text 'Web画面上で明細行を好きなだけ追加！' (Add as many detail rows as you like on the web screen!).

Web画面上で明細行を好きなだけ追加！

商品コード	商品名	数量	単価	金額	
120149	商品	0	1000	0	削除
		0	0	0	削除

# ■ ここまでやりたい！本格Webアプリケーション開発

## 1.動的な明細画面の実装

### フレームを使った明細の動的制御

フレームとは・・・

画面(フォーム)をコンポーネントのように部品化できる機能

### ファイル>新規作成>その他

①メニューの  
[ファイル|新規作成|その他]を実行。

②new Frameを選ぶとフォームと同じ  
形式で部品が作成できる。

# ■ ここまでやりたい！本格Webアプリケーション開発

## 1.動的な明細画面の実装

### フレームを使った明細の動的制御

フレームで明細行のテンプレートを開発する。

#### フレーム

商品コード	商品名	数量	単価	金額	
<input type="text"/>	<input type="text" value="sqlqrySHNMST"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="button" value="削除"/>

明細の1行分をフレームとして作成する

注文明細					
商品コード	商品名	数量	単価	金額	
<input type="text"/>	<input type="text"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="button" value="削除"/>

# ■ ここまでやりたい！本格Webアプリケーション開発

## 1.動的な明細画面の実装

### フレームを使った明細の動的制御

明細追加ボタンを押下した際に、動的にフレームを追加する。

商品コード	商品名	数量	単価	金額	
<input type="text"/>	sqlqrySHNMST	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	削除

追加ボタンで動的にフレームをコピー作成。

明細追加

注文確認

注文明細

商品コード	商品名	数量	単価	金額	
<input type="text"/>	<input type="text"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	削除
<input type="text"/>	<input type="text"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	削除
<input type="text"/>	<input type="text"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	削除

フレームはTIWRegionの上に作成する。  
※ TIWRegionはWebでのパネルのような部品。

# ■ ここまでやりたい！本格Webアプリケーション開発

## 1.動的な明細画面の実装

### フレームを使った明細の動的制御

#### 明細追加ボタンの処理(画面側ソース)

```
procedure TEntryfrm. IWButtonAddClick(Sender: TObject);
var
  Frame : TFrame1;    //明細用のフレーム
begin
  Inc(FframeCnt);    //追加するフレームの番号をカウント
  Frame      := TFrame1.Create(reMeisai); //フレームの作成
  Frame.Parent := IWRegion1;           //IWRegionを親(土台)に設定
  //作成したフレームを制御できるようにカウントした番号で名前を設定
  Frame.Name  := 'Frame' + FormatFloat('00', FframeCnt);
end;
```

# ■ ここまでやりたい！本格Webアプリケーション開発

## 1.動的な明細画面の実装

### フレームを使った明細の動的制御

#### 明細フレームの削除ボタンの処理(フレーム側ソース)

```
procedure TFrame1. IWButtonDeleteClick(Sender: TObject);  
begin  
    Self.Free; //明細で削除ボタンを押下したらフレーム自身で解放  
end;
```

# ■ ここまでやりたい！本格Webアプリケーション開発

## 1.動的な明細画面の実装

ログイン中  
株式会社ミガロ様  
ログアウト

注文入力画面

会社名: 株式会社ミガロ.  
ご担当者: 吉原 泰介  
納入先郵便番号: 556-0017  
納入先住所: 大阪市浪速区湊町2-1-57難波サンケイビル 13F  
納入先電話番号: 06-6251-8601 希望納品日: 2012/11/21

合計金額 ¥1,050

明細追加 注文確認

商品コード	商品名	数量	単価	金額	
120149	商品	0	1000	0	削除
		0	0	0	削除
		0	0	0	削除

Web画面上で明細行を好きなだけ追加！

削除

Web画面上で不要な明細行は削除！

# ■ ここまでやりたい！本格Webアプリケーション開発

## 機能にこだわる本格Webアプリケーション！

### 2.ポップアップ画面の実装

# ■ ここまでやりたい！本格Webアプリケーション開発

## 2.ポップアップ画面の実装

Webアプリケーションでの補助検索等の画面は、画面遷移で実装することが多い。

The screenshot shows a web browser window with the URL `http://win-916fd1e4dvd/DEMO/DEMOAPR.dll/`. The main page displays a table for adding products. A search popup window is overlaid on top, titled '製品検索' (Product Search). The popup contains search criteria and a list of products.

**製品検索** 型式、名称、ベルト幅から製品を検索します。

型式  前方一致  
名称  部分一致

(mm)  
◎在庫あり △要問合せ 一委注生産品 ☆組合せ品 (要問合せ)

検索

1件目～8件目 / 8件中

型式	製品名称	在庫	
MGR-0001	<a href="#">工具セットA KA-0010</a>	—	<a href="#">カートに追加</a>
MGR-0005	<a href="#">ネジ NJ-00360</a>	△	<a href="#">カートに追加</a>
MGR-0003	<a href="#">電動ドリル DD-0085</a>	—	<a href="#">カートに追加</a>
MGR-0004	<a href="#">ミガロ.OR MGR-0220</a>	☆	<a href="#">カートに追加</a>
MGR-0006	<a href="#">ネジ保護カバー NH-00390</a>	◎	<a href="#">カートに追加</a>
MGR-0002	<a href="#">工具セットB KB-385</a>	☆	<a href="#">カートに追加</a>
MGR-0007	<a href="#">ドライバー DR-00620</a>	—	<a href="#">カートに追加</a>
MGR-0008	<a href="#">特殊A TA-0500</a>	—	<a href="#">カートに追加</a>

# ■ ここまでやりたい！本格Webアプリケーション開発

## 2.ポップアップ画面の実装

ポップアップ画面を別DLLとして作成してJavaScriptを組み込むことで、別ブラウザのポップアップ画面連携が実現できる！

検索ボタンからポップアップ画面起動！

別ブラウザと連携

商品コード	商品名	価格	
120149	エアフォート	16,920	選択
120150	SFチェアー	38,680	選択
120157	書棚	26,500	選択
20156	セットデスク	38,200	選択
120155	木製長机	7,280	選択
120154	スチール机	26,800	選択
120153	6人用ロッカー	21,200	選択

# ■ ここまでやりたい！本格Webアプリケーション開発

## 2.ポップアップ画面の実装

### 2つのWeb画面を連携してポップアップ画面制御

C/Sアプリケーションと違い、別ブラウザでポップアップするサブ画面は別のアプリケーション(DLL)として作成する必要がある。  
ただし、通常ブラウザ間では値の受渡を行うことはできない。

#### 親アプリケーション

The screenshot shows the main application window titled "MIGARO. Technical Seminar". The user is logged in as "株式会社ミガロ様". The page is divided into a header, a login section, and a main content area. The main content area is titled "注文入力画面" (Order Entry Screen) and contains a form for entering order details. The form includes fields for company name, contact person, postal code, address, and phone number. There are also buttons for "明細追加" (Add Detail) and "注文確認" (Confirm Order). A summary box shows the order total (¥0), shipping fee (¥1,050), and total amount (¥1,050). Below the form is a table for order details.

商品コード	商品名	数量	単価	金額	
		0	0	0	削除
		0	0	0	削除

#### 子アプリケーション

The screenshot shows a sub-application window titled "商品選択一覧" (Product Selection List). It displays a table of products with columns for product code, product name, price, and a selection button. The products listed are:

商品コード	商品名	価格	
120149	エアフォート	16,920	選択
120150	SFチェア	38,680	選択
120157	書棚	26,500	選択
120156	セットデスク	38,200	選択
120155	木製長机	7,280	選択
120154	スチール机	26,800	選択
120153	6人用ロッカー	21,200	選択

# ■ ここまでやりたい！本格Webアプリケーション開発

## 2.ポップアップ画面の実装

### 2つのWeb画面を連携してポップアップ画面制御

#### ！ポイント

JavaScriptでブラウザの親子関係を利用すれば親/子のアプリケーション間で値を受け渡すことができる。

### 親アプリケーション

ログイン中  
株式会社ミガロ様  
ログアウト

MIGARO Technical Seminar

注文入力画面

会社名 株式会社ミガロ  
ご担当者 吉原 泰介  
納入先郵便番号 556-0017  
納入先住所 大阪市浪速区湊町2-1-57難波サンビル13F  
納入先電話番号 06-6251-8601 希望納品日 2012/11/21

注文明細金額 ￥0  
送料 ￥1,050  
合計金額 ￥1,050

商品コード	商品名	数量
		0

### 子アプリケーション

商品コード	商品名	価格	
120149	エアフォート	16,920	選択
120150	SFチェア	38,680	選択
120157	書棚	26,500	選択
120156	セットデスク	38,200	選択
120155	木製長机	7,280	選択
120154	スチール机	26,800	選択
120153	6人用ロッカー	21,200	選択

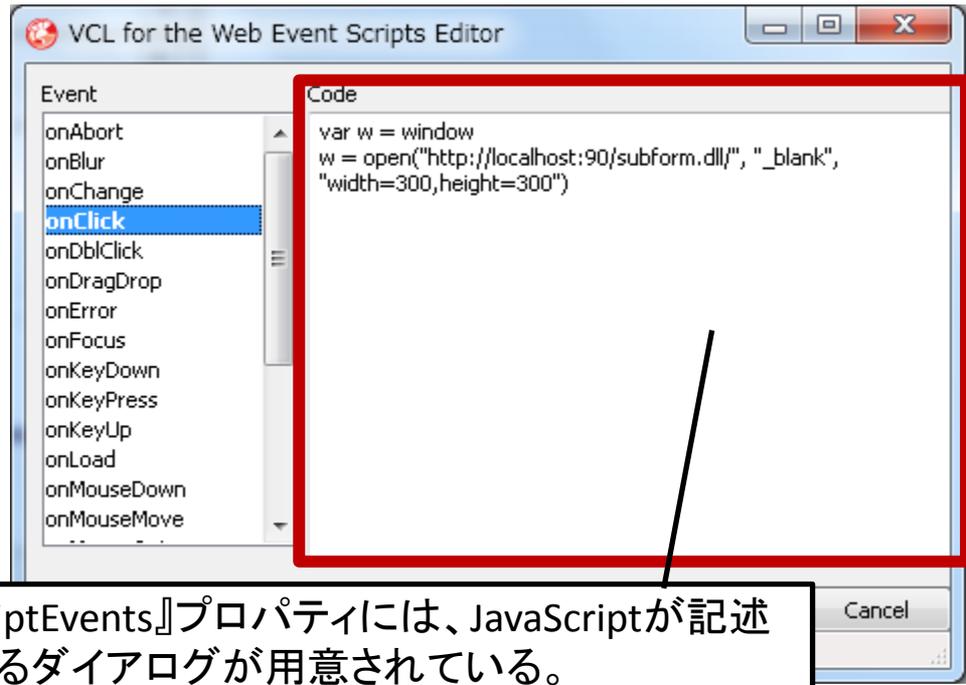
JavaScriptでは  
起動元画面を操作可能

# ■ ここまでやりたい！本格Webアプリケーション開発

## 2.ポップアップ画面の実装

### 2つのWeb画面を連携してポップアップ画面制御

Delphi/400でJavaScriptを記述するには専用のプロパティが用意されている。



Webのコンポーネントには『ScriptEvents』プロパティが用意されている。

『ScriptEvents』プロパティには、JavaScriptが記述できるダイアログが用意されている。

# ■ ここまでやりたい！本格Webアプリケーション開発

## 2.ポップアップ画面の実装

### 2つのWeb画面を連携してポップアップ画面制御

子アプリケーション起動時のJavaScript

### 検索ボタンのJavaScript処理(親アプリケーション側ソース)

// 『ScriptEvents』のOnClickイベント

```
var w = window
```

```
w = open( "http://Webサーバ/subform.dll/", "_blank", "width=300,height=300")
```

子アプリケーションを新しいWindowで起動。

親アプリケーション

注文明細

商品コード	商品名
<input type="text"/>	<input type="text"/>

起動

子アプリケーション

商品選択一覧

商品コード	商品名	価格	選択
120140	ガラスジョード	16,900	選択
120150	ガラスジョード	36,000	選択
120157	ステンレス製	46,500	選択
120156	ステンレス製	6,200	選択
120155	ステンレス製	7,200	選択
120154	ステンレス製	86,000	選択
120153	ステンレス製	21,200	選択

# ■ ここまでやりたい！本格Webアプリケーション開発

## 2.ポップアップ画面の実装

### 2つのWeb画面を連携してポップアップ画面制御

親アプリケーションに結果を返すJavaScript

#### 決定ボタンのJavaScript処理(子アプリケーション側ソース)

// 『ScriptEvents』のOnClickイベント

```
window.opener.document.forms[0].IWEDIT1.value  
= document.forms[0].IWEDIT2.value;  
window.close();
```

子アプリケーションの値を  
Opener(親アプリケーション)の項目にセットする。  
※Delphi/400のコンポーネントのhtml名も利用可能。

値をセットしたら子アプリケーション終了

#### 子アプリケーション

商品コード	商品名	価格	選択
120149	アウターウェア	16,900	<input type="checkbox"/>
120150	アルミワイヤ製網	36,600	<input type="checkbox"/>
120157	ステール製網	46,500	<input type="checkbox"/>
120169	ステンレス製網	6,200	<input type="checkbox"/>
120155	製網	7,200	<input type="checkbox"/>
120154	サンゴ石	96,800	<input type="checkbox"/>
120153	アルミ製網(15L)	21,200	<input type="checkbox"/>

結果をセット

#### 親アプリケーション

商品コード	商品名	数量
<input type="text" value="120149"/>	<input type="text" value="商品"/>	<input type="text" value="0"/>

# ■ ここまでやりたい！ 本格Webアプリケーション開発

## 2.ポップアップ画面の実装

別ブラウザ間の連携が実現！

The image shows a web browser window displaying a 'MIGARO. Technical Seminar' website. The main page is titled '注文入力画面' (Order Entry Screen) and contains a form for entering order details. A red box highlights the search fields for '商品コード' (Product Code) and '商品名' (Product Name). A red arrow points from these fields to a pop-up window titled '商品選択一覧' (Product Selection List). The pop-up window displays a table of products with columns for '商品コード' (Product Code), '商品名' (Product Name), and '価格' (Price). A text box at the bottom of the pop-up window states: 'C/Sアプリケーションのように値を受け渡せる！' (Can pass values like a C/S application!).

商品コード	商品名	価格	選択
120149	エアフォート	16,920	選択
120150	SFチェアー	38,680	選択
120151	書棚	26,500	選択
120152	セットデスク	38,200	選択
120155	木製長机	7,280	選択
120153	6人用ロッカー	21,200	選択

# ■ ここまでやりたい！本格Webアプリケーション開発

## 機能にこだわる本格Webアプリケーション！

### 3.PDF出力機能の実装

会社名

ご担当者

納入先郵便番号

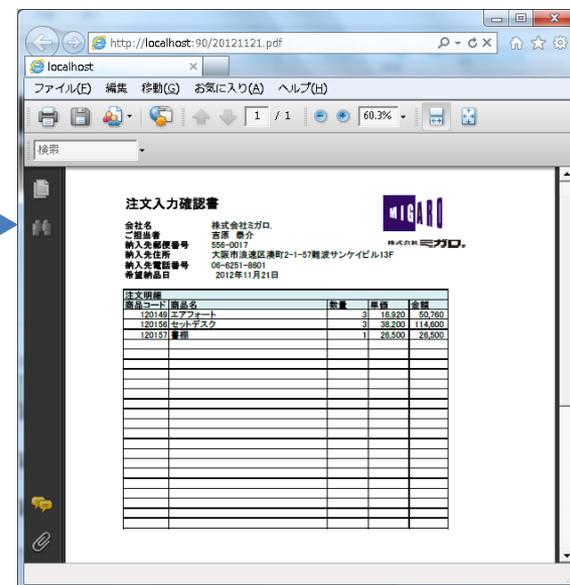
納入先住所

納入先電話番号  希望納品日

注文明細金額	¥ 191,860
送料	¥ 1,050
合計金額	¥ 192,910

明細追加 注文確認 PDF出力

商品コード	商品名	数量	単価	金額	
<input type="text" value="120149"/>	<input type="text" value="エアフォート"/>	<input type="text" value="3"/>	<input type="text" value="16,920"/>	<input type="text" value="50,760"/>	<input type="button" value="削除"/>
<input type="text" value="120156"/>	<input type="text" value="セットデスク"/>	<input type="text" value="3"/>	<input type="text" value="38,200"/>	<input type="text" value="114,600"/>	<input type="button" value="削除"/>
<input type="text" value="120157"/>	<input type="text" value="書棚"/>	<input type="text" value="1"/>	<input type="text" value="26,500"/>	<input type="text" value="26,500"/>	<input type="button" value="削除"/>



・・・PDF実装方法はお持ち帰り資料に詳しく掲載

## ■ ここまでやりたい！本格Webアプリケーション開発

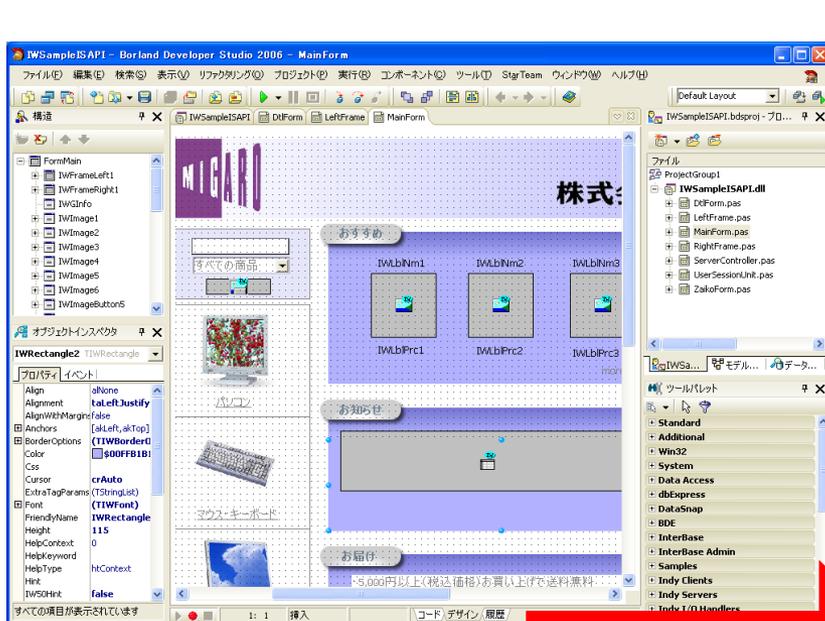
画面にこだわる本格Webアプリケーション！

4.htmlを連携したWeb画面実装

# ■ ここまでやりたい！ 本格Webアプリケーション開発

## 4.htmlを連携したWeb画面実装

VCL for the Webではフォームにコンポーネントを配置してC/Sアプリと同じようにWeb画面を作成できる。



コンパイル

# ■ ここまでやりたい！本格Webアプリケーション開発

## 4.htmlを連携したWeb画面実装

Web画面はコンポーネントで簡単に作成できるけど・・・



コンポーネントの見た目を変えたい・・・

ボタン

作りたい画面デザインのhtmlなら見つけたのに・・・



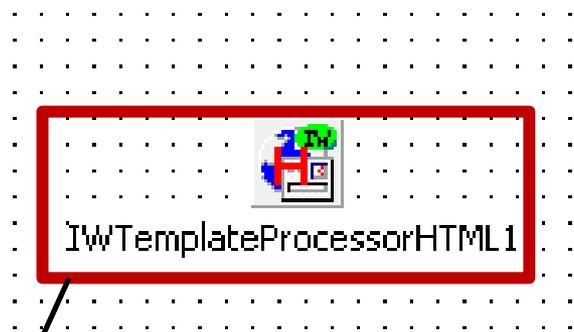
IWTemplateProcessorHTMLコンポーネントで解決！

# ■ ここまでやりたい！本格Webアプリケーション開発

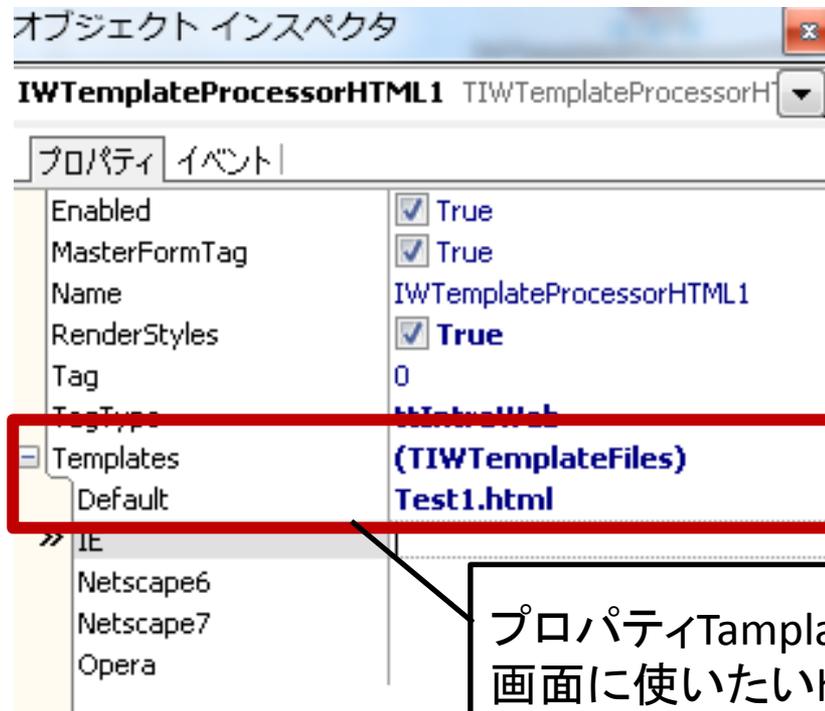
## 4.htmlを連携したWeb画面実装

### htmlを画面に連携できる便利なコンポーネント

#### IWTemplateProcessorHTMLコンポーネントの使い方①



IWTemplateProcessorHTML



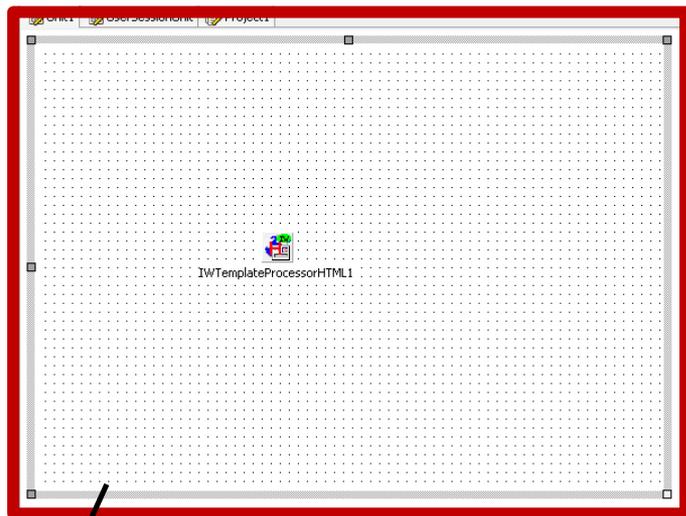
プロパティTemplates:  
画面に使いたいhtml名を設定  
※ブラウザ毎に設定も可能

# ■ ここまでやりたい！本格Webアプリケーション開発

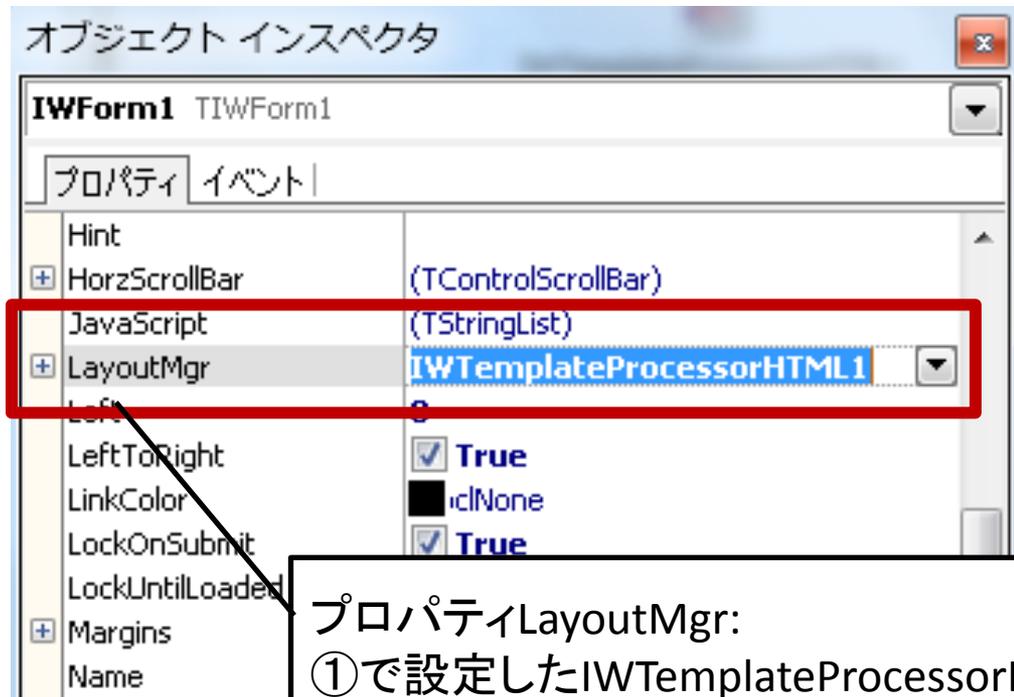
## 4.htmlを連携したWeb画面実装

### htmlを画面に連携できる便利なコンポーネント

#### IWTemplateProcessorHTMLコンポーネントの使い方②



Form(画面フォーム)



プロパティLayoutMgr:  
①で設定したIWTemplateProcessorHTML  
を設定

# ■ ここまでやりたい！本格Webアプリケーション開発

## 4.htmlを連携したWeb画面実装

### htmlを画面に連携できる便利なコンポーネント

#### IWTemplateProcessorHTMLコンポーネントの使い方③



アプリケーションを実行するフォルダ内に『templates』という名前で作成する

フォルダ内に①で指定したhtmlファイルを配置する

### ポイント

Delphi/400 V2009以降ではhtmlファイルは必ずUTF-8で保存

文字コード(E): UTF-8

保存(S)

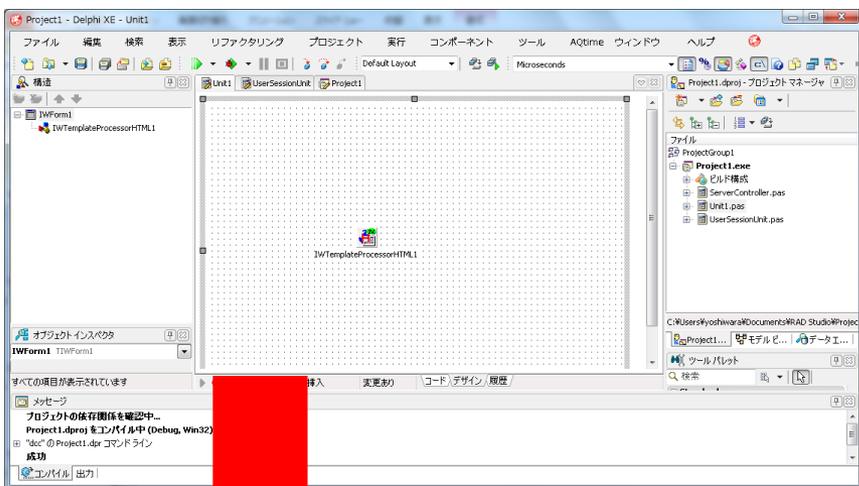
キャンセル

# ■ ここまでやりたい！本格Webアプリケーション開発

## 4.htmlを連携したWeb画面実装

### htmlを画面に連携できる便利なコンポーネント

#### IWTemplateProcessorHTMLコンポーネントの使い方④



Delphi/400で作成していない画面がhtmlから読み込まれた！

# ■ ここまでやりたい！ 本格Webアプリケーション開発

## 4.htmlを連携したWeb画面実装

### htmlを画面に連携できる便利なコンポーネント

#### IWTemplateProcessorHTMLコンポーネントの使い方⑤

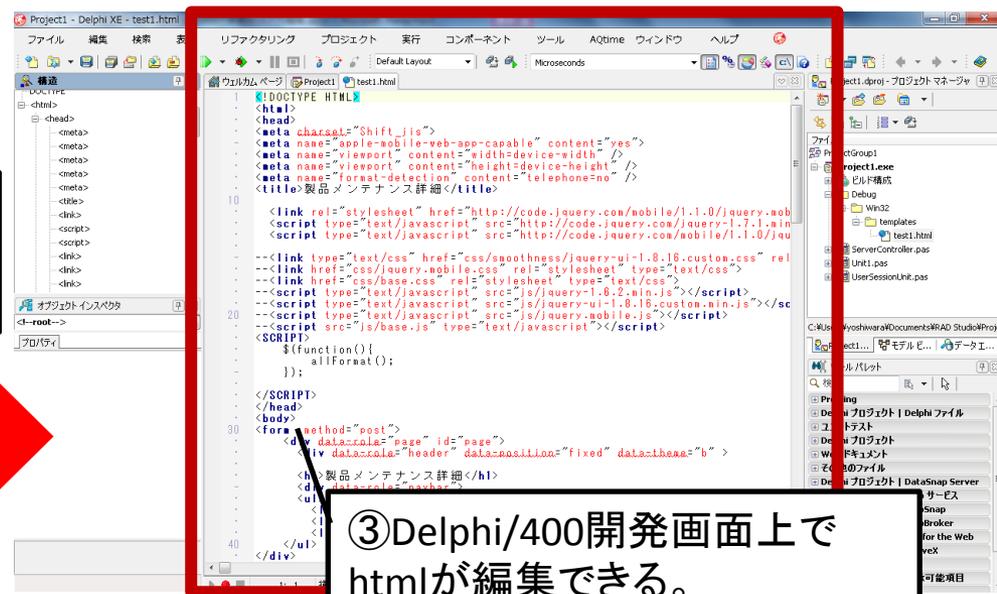
### htmlソースをDelphi/400の開発画面で編集する



①メニューの[プロジェクト|プロジェクトに追加]を実行。



②ダイアログからhtmlを選択。



③Delphi/400開発画面上でhtmlが編集できる。

# ■ ここまでやりたい！ 本格Webアプリケーション開発

## 4.htmlを連携したWeb画面実装

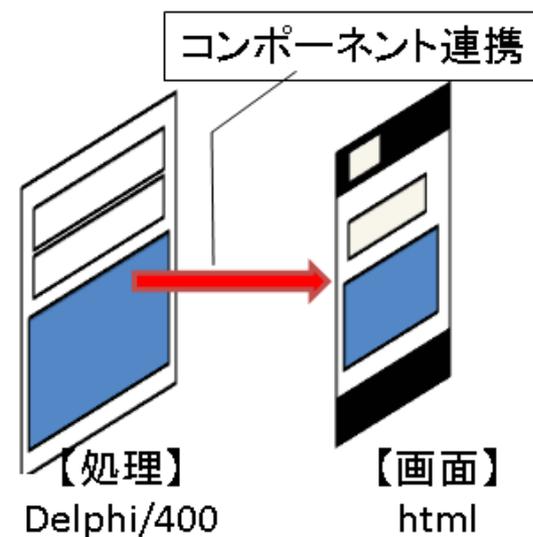
### htmlを画面に連携できる便利なコンポーネント

#### IWTemplateProcessorHTMLコンポーネントの使い方⑥

### htmlにDelphi/400のコンポーネントを組み込む

html画面でDelphi/400のコンポーネントを組み込むにはhtmlにコンポーネント名を {%コンポーネント名%} と記述しておく。

例) IWButton1をhtmlに組み込む場合 {%IWButton1%} と記述した部分にコンポーネントを連携して表示・操作できる。



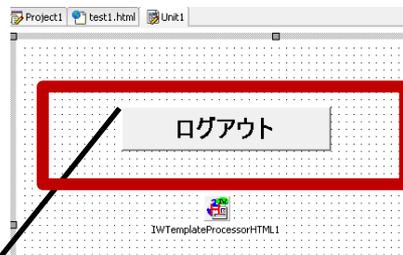
# ■ ここまでやりたい！本格Webアプリケーション開発

## 4.htmlを連携したWeb画面実装

### htmlを画面に連携できる便利なコンポーネント

#### IWTemplateProcessorHTMLコンポーネントの使い方⑦

### htmlにDelphi/400のコンポーネントを組み込む

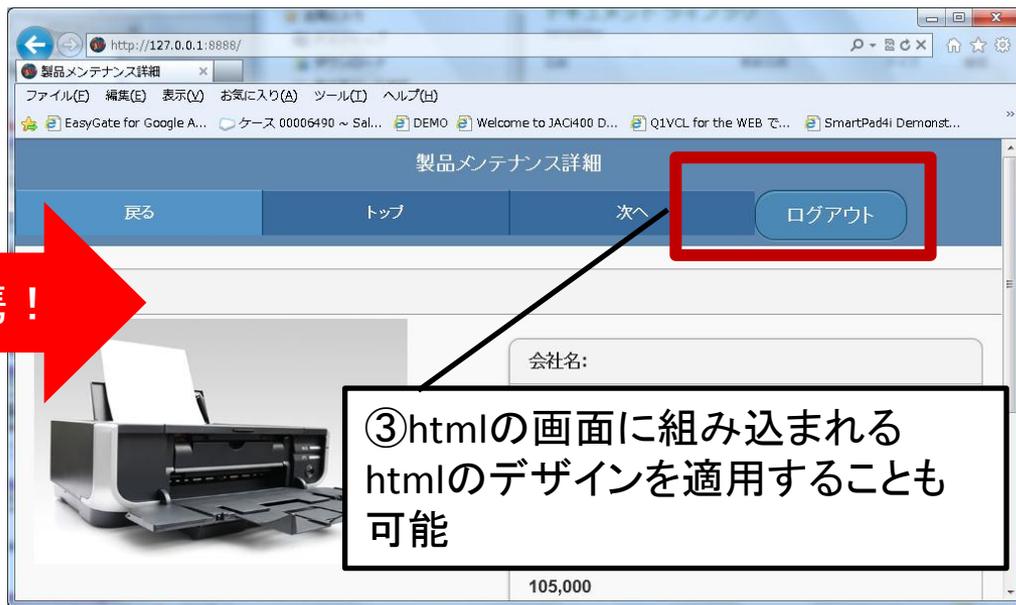


① Delphi/400のコンポーネント



② htmlにコンポーネント名を記述

連携！



③ htmlの画面に組み込まれるhtmlのデザインを適用することも可能

## ■ ここまでやりたい！本格Webアプリケーション開発

画面にこだわる本格Webアプリケーション！

5.jQueryMobileを利用したスマートデバイスWeb画面実装

# ■ ここまでやりたい！本格Webアプリケーション開発

## 5.jQueryMobileを利用したスマートデバイスWeb画面実装

### jQueryMobileのスマートデバイスWeb画面

スマートフォンやタブレット向けにこだわったWeb画面を作成するには『jQueryMobile』を利用すると便利。



# ■ ここまでやりたい！本格Webアプリケーション開発

## 5.jQueryMobileを利用したスマートデバイスWeb画面実装

### jQueryMobileのスマートデバイスWeb画面

jQueryMobileとは・・・

スマートフォンやタブレットで使用するWebアプリケーション用に画面デザインや部品を提供してくれるオープンソース

#### 公式サイト



<http://jquerymobile.com/>

#### 日本語リファレンス



[http://dev.screw-axis.com/doc/jquery\\_mobile/](http://dev.screw-axis.com/doc/jquery_mobile/)

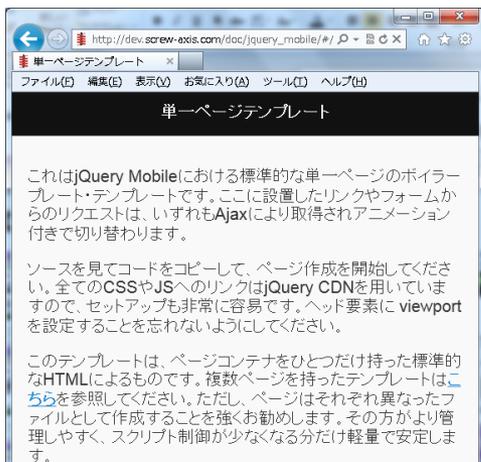
# ■ ここまでやりたい！ 本格Webアプリケーション開発

## 5. jQueryMobileを利用したスマートデバイスWeb画面実装

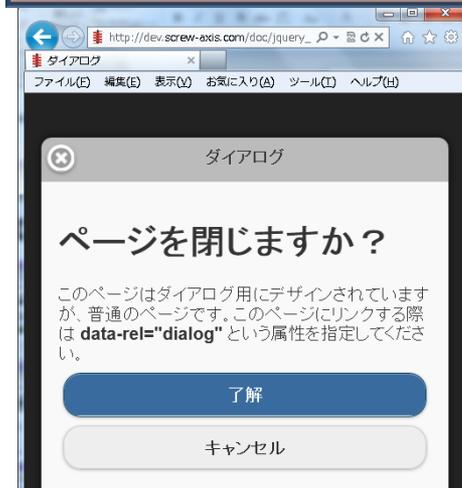
### jQueryMobileのスマートデバイスWeb画面

jQueryMobileで用意されているテンプレート(一例)

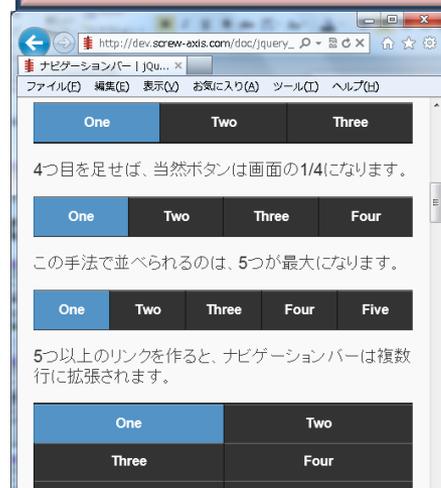
#### ページデザイン



#### ダイアログ



#### ツールバー



#### リストビュー



これ以外にも、よく見るスマートデバイス風画面、部品がhtmlソースでたくさん提供されている。

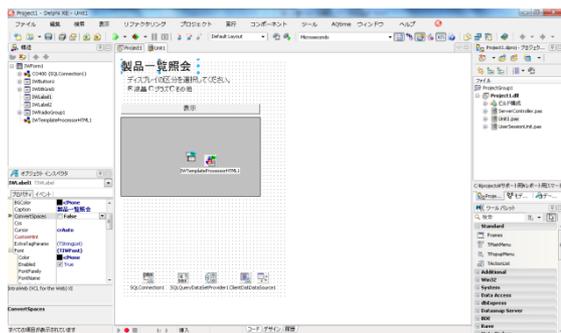
# ■ ここまでやりたい！本格Webアプリケーション開発

## 5.jQueryMobileを利用したスマートデバイスWeb画面実装

### Delphi/400とjQueryMobileの連携

jQueryMobileに用意されている画面テンプレートを  
IWTemplateProcessorHTMLコンポーネントで利用すれば  
簡単に本格的なスマートデバイス画面デザインが実現！

Delphi/400



jQueryMobile



+

=



格好が良い見た目だけでなく、  
画面サイズの自動調整なども実装  
されているので便利！

# ■ ここまでやりたい！本格Webアプリケーション開発

## 5.jQueryMobileを利用したスマートデバイスWeb画面実装

### Delphi/400とjQueryMobileの連携

基本は画面をjQueryMobileで利用して、ボタンや入力などをDelphi/400のコンポーネントで連携すればOK！

#### ！ポイント

コンポーネントにないjQueryMobile独自の部品を連携したい場合はどうすれば良いのだろうか？



IWDBGridではなくて  
リストビューで商品一覧を  
表示したいなあ

例えばスマートデバイス風の  
リストビュー



# ■ ここまでやりたい！本格Webアプリケーション開発

## 5. jQueryMobileを利用したスマートデバイスWeb画面実装

### Delphi/400とjQueryMobileの連携

jQueryMobile独自の部品を使う場合には  
Delphi/400のプログラムでjQueryMobileのhtmlソースを  
編集して出力することができる。

htmlのソースの書き方は  
サイト内で詳しく記載されている。



#### リンクのリスト

リンクのリストは、シンプルに順序無しリスト(`ul`)にリンクアイテムを含め、属性で **data-role="listview"** を指定することで記述できます。jQuery Mobileはリストを画面の横幅いっぱいにし、右側に矢印を付けるなど必要なスタイル設定を行います。リストアイテムがタップされると、フレームワークは該当するアイテムの最初のリンク対象をAjaxで取得し、新しいページをDOM上にロードし、ページの切り替えを行います。リストビューに指定可能な属性については[data属性リファレンス](#)を参照してください。

```
<ul data-role="listview" data-theme="g">
  <li><a href="acura.html">Acura</a></li>
  <li><a href="audi.html">Audi</a></li>
  <li><a href="bmw.html">BMW</a></li>
</ul>
```

# ■ ここまでやりたい！本格Webアプリケーション開発

## 5.jQueryMobileを利用したスマートデバイスWeb画面実装

### Delphi/400とjQueryMobileの連携

htmlソースを出力する処理を作成する。

#### htmlソース作成処理(ソース) 例) ClientDataSetからリスト作成

##### 宣言部

```
public
```

```
function RenderListView: String;
```

```
...
```

##### 実装部

```
function TIWForm1.RenderListView: String;
```

```
var
```

```
ul : TStringList;
```

```
begin
```

```
//リストビューのhtmlソースを動的に作成していく。
```

```
ul := TStringList.Create;
```

```
ul.Add('<ul data-role="listview">');
```

htmlソースを作成する関数

StringListにhtmlソース文を格納していく

# ■ ここまでやりたい！本格Webアプリケーション開発

## 5.jQueryMobileを利用したスマートデバイスWeb画面実装

//ClientDataSetのデータ分、リストビューのレコードを作成する。

```
while not(ClientDataSet1.Eof) do
```

```
begin
```

```
    ul.Add('<li>');
```

```
    ul.Add('<a><img src= "/images/'
```

```
        + ClientDataSet1.FieldName('画像ファイルパス').AsString + '.png">');
```

```
    ul.Add('<h3>' + ClientDataSet1.FieldName('商品コード').AsString);
```

```
    ul.Add(ClientDataSet1.FieldName('商品名').AsString + '</h3>');
```

```
    ul.Add('<p>価格:' + ClientDataSet1.FieldName('価格').AsString + '</p>');
```

```
    ul.Add('<p>在庫:' + ClientDataSet1.FieldName('在庫数').AsString + '</p>');
```

```
    ul.Add('</a></li>');
```

```
    ClientDataSet1.Next;
```

```
end;
```

```
ul.Add('</ul>');
```

//作成したhtml分を関数から返却する。

```
Result := ul.Text;
```

```
FreeAndNil(ul);
```

```
end;
```

リストビューに表示したい  
項目(データや画像)をレコード  
を読み取りながらループで  
htmlソースを作成していく。

# ■ ここまでやりたい！ 本格Webアプリケーション開発

## 5. jQueryMobileを利用したスマートデバイスWeb画面実装

### Delphi/400とjQueryMobileの連携

htmlソースを出力する処理を呼び出すイベント。

The screenshot shows the Delphi Object Inspector window titled "オブジェクトインスペクタ". The selected object is "IWTemplateProcessorHTML1" of type "TIWTemplateProcessorHTML". The "イベント" (Events) tab is active, showing a list of events: "OnAfterProcess", "OnBeforeProcess", and "OnUnknownTag". The "OnUnknownTag" event is selected, and its handler is set to "IWTemplateProcessorHTML1UnknownTag". A red box highlights the "OnUnknownTag" event and its handler. A callout box points to this event with the text "htmlソースを追記する場合は OnUnknownTagイベントを利用". Another callout box points to the object name in the Object Inspector with the text "IWTemplateProcessorHTML".

オブジェクトインスペクタ

IWTemplateProcessorHTML1 TIWTemplateProcessorHTML

プロパティ イベント

イベント	ハンドラ
OnAfterProcess	
OnBeforeProcess	
OnUnknownTag	IWTemplateProcessorHTML1UnknownTag

htmlソースを追記する場合は OnUnknownTagイベントを利用

IWTemplateProcessorHTML1

IWTemplateProcessorHTML

# ■ ここまでやりたい！本格Webアプリケーション開発

## 5.jQueryMobileを利用したスマートデバイスWeb画面実装

### Delphi/400とjQueryMobileの連携

htmlソースを出力する処理を作成する。

#### htmlソース作成処理を呼び出す(ソース)

```
procedure TIWForm1.IWTemplateProcessorHTML1UnknownTag(const AName: string;
  var VValue: string);
begin
  //html上で{%ListView%}を描画する際に、データがオープンされていれば
  //リストビューを動的に作成する。
  if ((AName = 'ListView') and (ClientDataSet1.Active)) then
  begin
    VValue := RenderListView;
  end;
end;
```

htmlソースで『ListView』という  
部品を表示するときに  
作成したhtmlソースを出力する

# ■ ここまでやりたい！本格Webアプリケーション開発

## 5.jQueryMobileを利用したスマートデバイスWeb画面実装



htmlが編集されて完成！



## ■ 補足

### VCL for the Webを拡張する市販コンポーネント情報

- TMS IntaraWeb iPhone Controls Pack  
スマートデバイス向けWebアプリケーション用

<http://www.tmssoftware.com/site/tmsiiphone.asp>



- IWCGjQuery

<http://www.cgdevtools.com/>

デスクトップ向けWebアプリケーション用

<http://www.cgdevtools.com/?skill=iwcgjquery-suite>

スマートデバイス向けWebアプリケーション用

<http://www.cgdevtools.com/skill/iwcgjquery-mobile-suite/>



※ミガロ.がご提供している製品ではありません。



## ■ まとめ

- Webアプリケーションは様々な環境に対応できる。
- VCL for the WebはC/Sと同様に開発ができる。
- 明細はフレームで制御すれば動的に追加できる。
- ポップアップ画面はJavaScriptで受け渡しができる。
- IWTemplateProcessorHTMLでhtmlとの連携ができる。
- jQueryMobileでスマートデバイス画面も簡単に作れる。

ご清聴ありがとうございました。

## ■ お持ち帰り資料

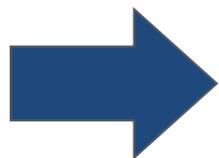
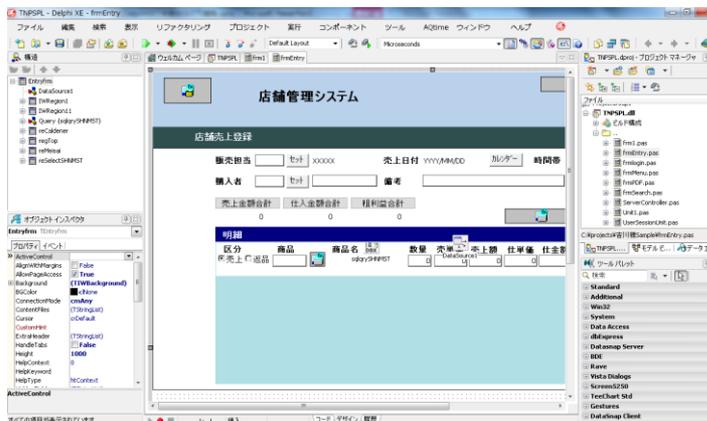
VCL for the Web基礎知識

# ■ お持ち帰り資料 (VCL for the Web 基礎知識)

## • VCL for the Web アプリケーションの開発手順

### スタンドアロン/アプリケーションモード手順

- ① 新規プロジェクトの作成
- ② コンポーネントを貼り付けてプロパティの設定
- ③ 必要に応じてイベントハンドラを作成
- ④ コンパイル
- ⑤ 実行(テスト)



C/Sアプリケーションと同じような手順で開発が可能

# ■ お持ち帰り資料 (VCL for the Web 基礎知識)

## • VCL for the Web プロジェクトの新規作成

### VCL for the Web プロジェクトの作成手順

1. ファイル⇒新規作成⇒その他 を選択
2. Delphiプロジェクト  
⇒VCL for the Web より、[VCL for the Web Application Wizard] を選択
3. ウィザード画面より作成モード、保存先、ファイル名等を指定

The image shows a sequence of three screenshots from the Delphi IDE illustrating the steps to create a VCL for the Web project. The first screenshot shows the 'New' dialog box with 'VCL for the Web' selected in the project categories. The second screenshot shows the 'VCL for the Web Application Wizard' dialog box with 'StandAlone Application' selected. The third screenshot shows the same wizard dialog box with 'Create User Session' checked and the project name and directory specified.

スタンドアロンモードの場合  
StandAlone Application  
アプリケーションモードの場合  
ISAPI Extension

Create User Sesseion にチェック  
(ユーザーセッションを使う)

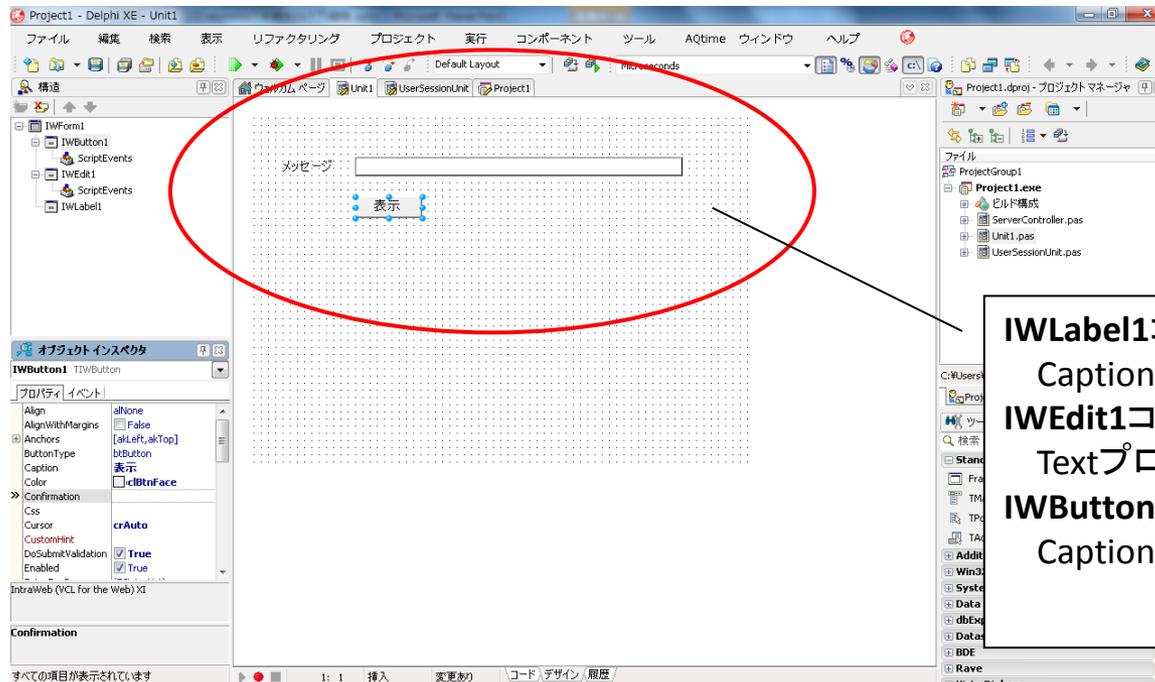
プロジェクトの保存先、  
プロジェクト名を指定

# ■ お持ち帰り資料 (VCL for the Web基礎知識)

## • コンポーネントの配置

### コンポーネントの貼り付けおよびプロパティの設定

1. ウィザードにて作成されたプロジェクトよりUnit1を選択し、フォームを表示
2. [IWStandard]ページより、IWLabel、IWEdit、IWButton をフォームに貼り付け プロパティをセット



**IWLabel1**コンポーネント  
Captionプロパティ = 'メッセージ'  
**IWEdit1**コンポーネント  
Textプロパティ = "" (空文字)  
**IWButton1**コンポーネント  
Captionプロパティ = '表示'

# ■ お持ち帰り資料 (VCL for the Web 基礎知識)

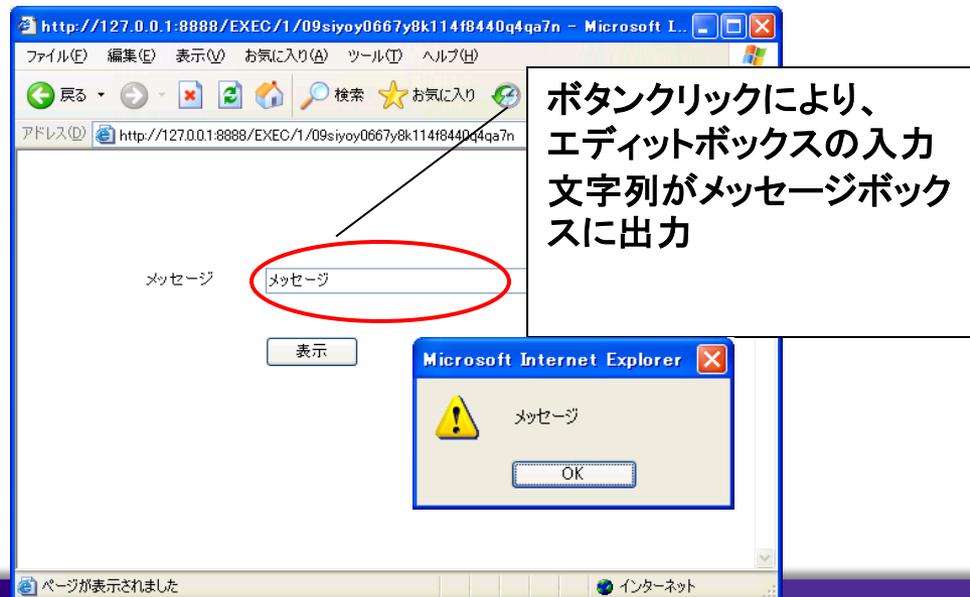
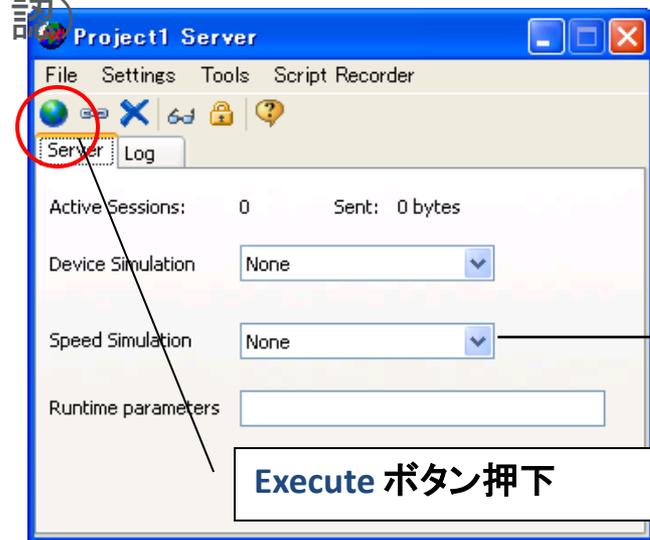
## • コンポーネントの配置

### イベントハンドラの設定とコンパイル/実行

1. IWButton1コンポーネントのonClickイベントハンドラを設定し、下記コードを入力

```
procedure TIWForm1.IWButton1Click(Sender: TObject);  
begin  
  WebApplication.ShowMessage(IWEdit1.Text);  
end;
```

2. コンパイル ⇒ 実行 (起動するフォームからブラウザ起動ボタンを押下し動作確認)



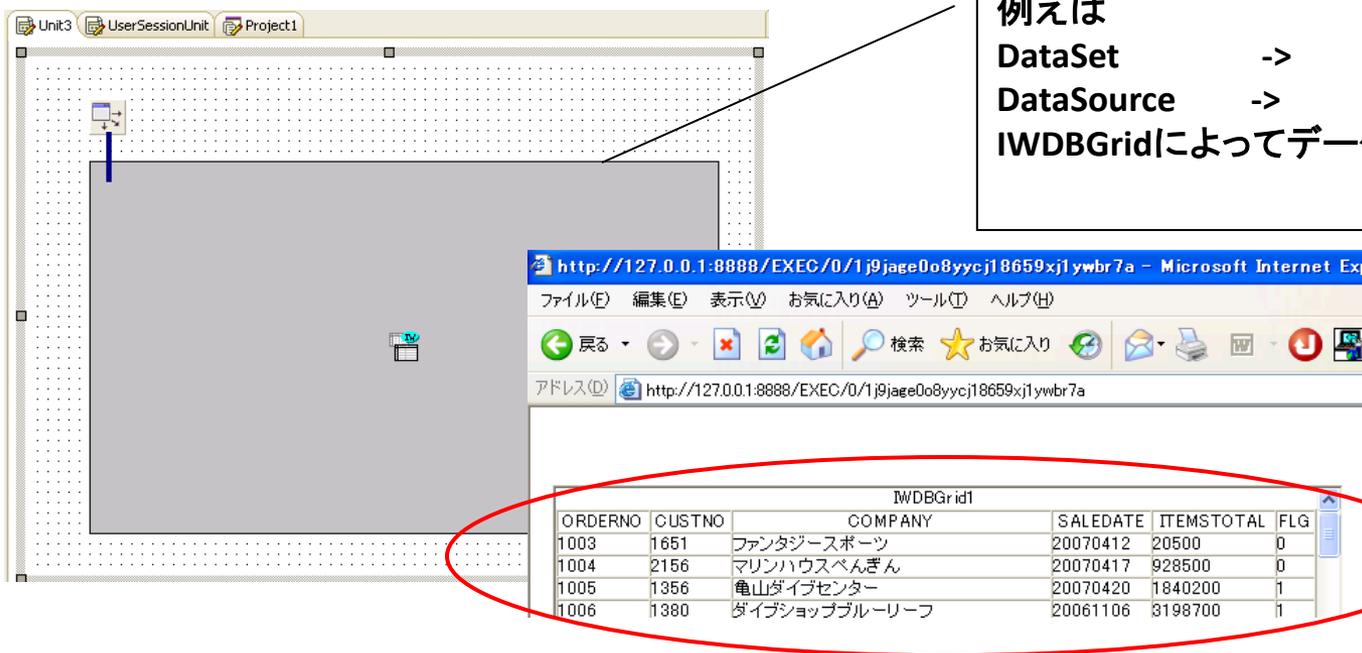
# ■ お持ち帰り資料 (VCL for the Web 基礎知識)

## • データベース接続

### DB連結用コンポーネント

- データベース接続はDBExpress等、GUIと同様に使用することができます。データベースに連結した画面を作成したい場合[IW Data]ページにコンポーネントが用意されています。

([Data Control]ページのコンポーネントに相当)



The screenshot shows a VCL IDE window with a web browser. The browser displays a table of data. A red oval highlights the table data.

例えば  
DataSet ->  
DataSource ->  
IWDBGridによってデータを一覧表示

ORDERNO	CUSTNO	COMPANY	SALEDATE	ITEMSTOTAL	FLG
1003	1651	ファンタジースポーツ	20070412	20500	0
1004	2156	マリンハウスべんぎん	20070417	928500	0
1005	1356	亀山ダイブセンター	20070420	1840200	1
1006	1380	ダイブショップブルーリーフ	20061106	3198700	1

# ■ お持ち帰り資料 (VCL for the Web 基礎知識)

## • 画面遷移

### 画面の遷移方法

- Delphiプロジェクト⇒ VCL for the Web より、[New Form] を選択して追加
- 次画面呼び出し元の処理
  - **Createメソッド**でオブジェクトを生成
  - **Showメソッド**で遷移
- 次画面の処理
  - 画面終了時に**Releaseメソッド**呼び出しにより、元のフォームに戻る

```
procedure TfrmMain.btnGoFrm2Click(Sender: TObject);  
var  
    frmDetail: TfrmDetail;  
begin  
    frmDetail := TfrmDetail.Create(WebApplication);  
    frmDetail.Show;  
end;
```

フォーム変数はGlobal定義されていないので、ローカル変数として定義

```
procedure TfrmDetail.btnBackClick(Sender: TObject);  
begin  
    Release;  
end;
```

# ■ お持ち帰り資料 (VCL for the Web基礎知識)

## • グローバル変数 グローバル変数の扱い

- WEBアプリケーションの場合、**WEBブラウザ単位に変数を保持しなければならない。**
- VCL for the Webの場合、UserSessionユニットのIWUserSessionクラスに変数を定義し、**UserSessionオブジェクトとして利用する。**

```
TIWUserSession = class(TIWUserSessionBase)
```

```
  sesSession: TSession;
```

```
  dbAS: TDatabase;
```

```
  qrySelectData: TQuery;
```

```
private
```

```
  { Private declarations }
```

```
public
```

```
  { Public declarations }
```

```
  LogOnUserID: String;
```

```
  LogOnUserName: String;
```

```
end;
```

グローバル変数  
WEBブラウザ毎に変  
数が別々に保持され  
る

```
// ログオンユーザーID
```

```
// ログオンユーザー名
```

```
procedure TfrmMain.IWAppFormRender(Sender: TObject);
```

```
begin
```

```
  // ログオンユーザー名の表示
```

```
  lblLogOnUser.Caption := UserSession.LogOnUserName;
```

```
end;
```

# ■ お持ち帰り資料

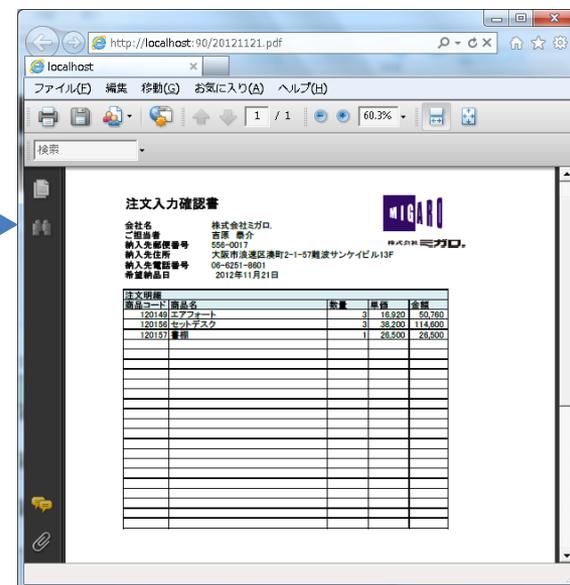
## Webアプリケーションでの PDF出力方法

※テクニカルレポートNo.3「Delphi/400 WEBからのPDF出力」にも詳しく掲載しています。

会社名	株式会社ミガロ	注文明細金額	¥ 191,860
ご担当者	吉原 泰介	送料	¥ 1,050
納入先郵便番号	556-0017	合計金額	¥ 192,910
納入先住所	大阪市浪速区湊町2-1-57難波サンケイビル13F		
納入先電話番号	06-6251-8601	希望納品日	2012/11/21

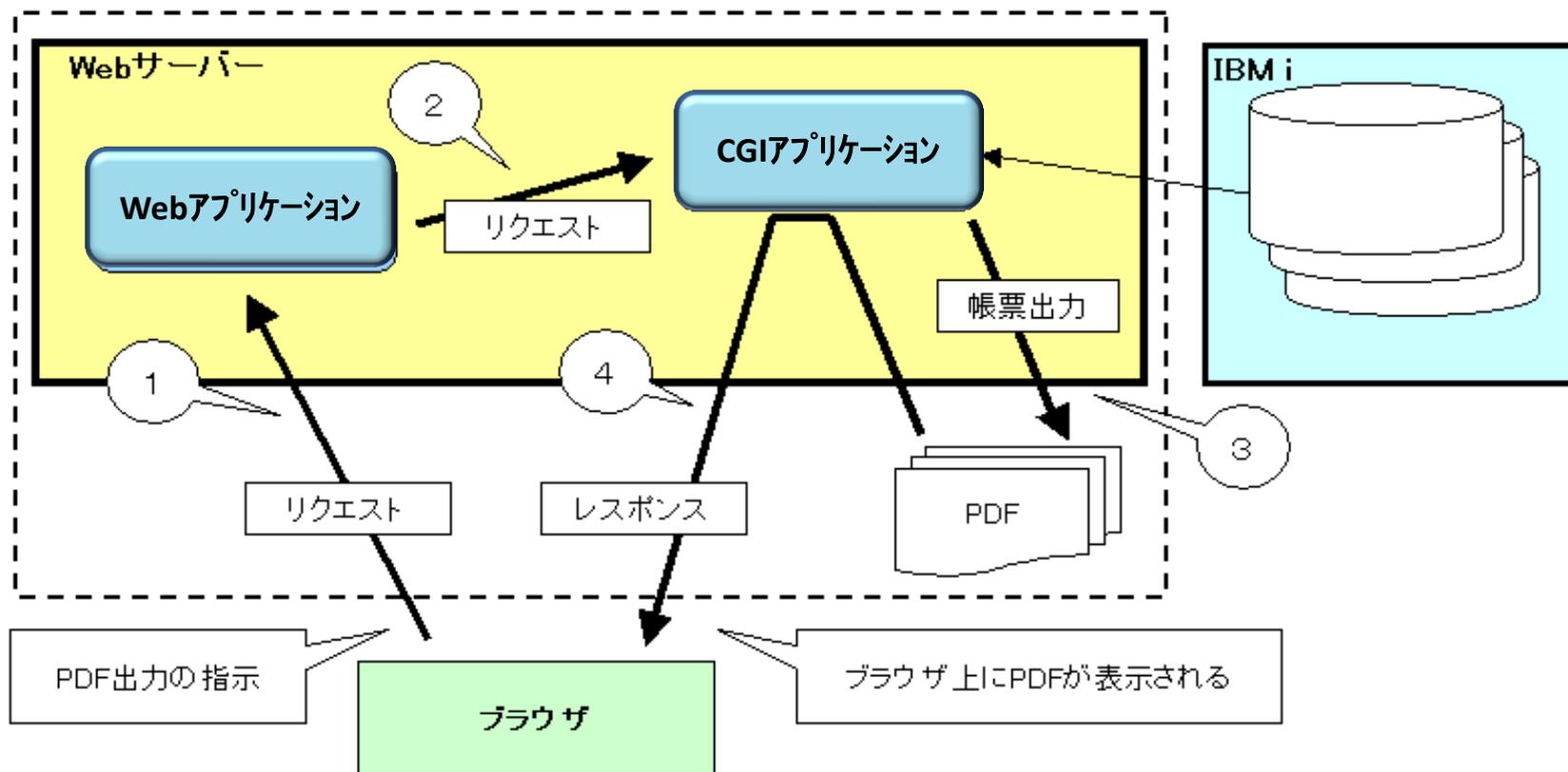
明細追加 注文確認 PDF出力

商品コード	商品名	数量	単価	金額	
120149	エアフォート	3	16,920	50,760	削除
120156	セットデスク	3	38,200	114,600	削除
120157	書棚	1	26,500	26,500	削除



## ■ お持ち帰り資料 (Webアプリでの PDF出力方法)

- WebアプリケーションでPDFを出力する仕組み  
PDFを出力するプログラムはCGIアプリケーションとして別途開発し、Webアプリケーション (VCL for the Web) から呼び出して利用する。

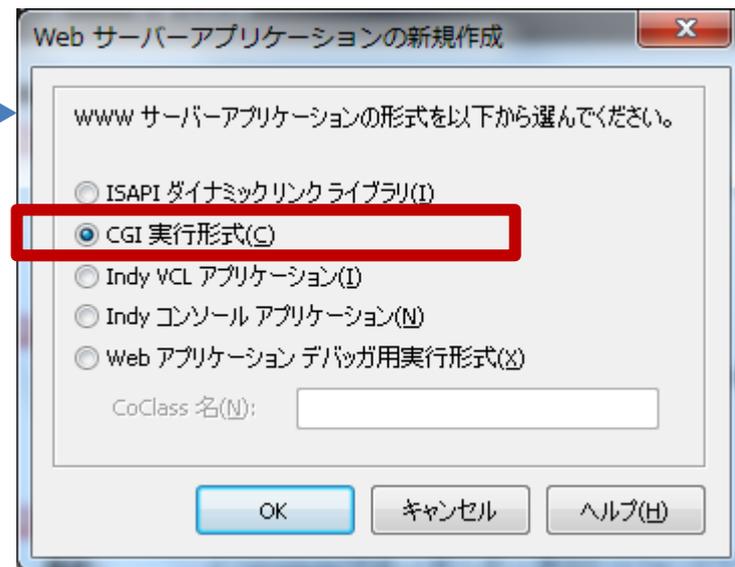


# ■ お持ち帰り資料 (Webアプリでの PDF出力方法)

## • CGIアプリケーションの作成方法①

### WebBrokerプロジェクト(CGI)の作成手順

1. ファイル⇒新規作成⇒その他 を選択
2. Delphiプロジェクト  
⇒WebBroker より、[Webサーバアプリケーション] を選択
3. アプリケーション形式を『CGI実行形式』を指定



# ■ お持ち帰り資料 (Webアプリでの PDF出力方法)

## • CGIアプリケーションの作成方法②

プロジェクトが作成されるとTWebModuleが用意されているので Actionsプロパティを選択して、Itemを作成し、呼び出されるURL部(PathInfo)にあわせて OnActionイベントにPDFを出力するプログラムを作成する。



The screenshot shows the 'WebModule1.Actions' editor. A table lists actions with their names, PathInfo, and Enabled status. A red box highlights the '/getPDF' PathInfo for 'WebActionItem1'. A callout box explains that PathInfo is the access path of the URL used to call the CGI. A diagram below shows the breakdown of a URL: 'http://migarco.jp/project1.exe/getpdf?code=0', with 'Pathinfo' highlighted in a red box and linked to the '/getpdf' value in the table.

Name	PathInfo	Enabled
DefaultHandler	/	True
WebActionItem1	/getPDF	True

PathInfoはCGIを呼び出す際のURLのアクセスパスになる。Actionのリストは右クリックから簡単に追加できるので、アクセスパスによって複数の機能を実装することも可能。

CGIを呼び出すURL例

schem host ScriptName Pathinfo Query  
http://migarco.jp/project1.exe/getpdf?code=0

# ■ お持ち帰り資料 (Webアプリでの PDF出力方法)

## • CGIアプリケーションの作成方法③

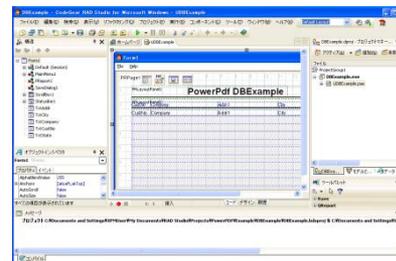
### PDFを作成するOnAction(ソース)

```
procedure TWebModule1.WebModule1WebActionItem1Action(Sender: TObject;  
  Request: TWebRequest; Response: TWebResponse; var Handled: Boolean);  
var  
  sFileName :String;  
  sFileStream : TFileStream; //返却用  
begin  
  //テンプレートファイルを生成  
  frmPrintPDF := TfrmPrintPDF.Create(nil);  
  
  try  
    //リクエスト (パスのQuery部) をパラメータにデータ抽出  
    SQLQuery1.ParamByName('CODE').AsString :=  
      Request.QueryFields.Values['CODE'];  
    SQLQuery1.Open;  
  
    //PDFの作成 (PowerPDF等)  
    with frmPrintPDF do  
      begin  
  
        //ここで抽出データからPDF帳票を作成する。  
  
      end;  
      PReport1.FileName := sFileName;
```

schem host ScriptName Pathinfo Query  
http://migaro.co.jp/project1.exe/getpdf?code=0

呼び出し時のパラメータ (Query部) を抽出に使うこともできる。

PDFの作成はPowerPDFなどでの開発が便利。  
※詳しくはテクニカルレポートNo3  
「Delphi/400帳票開発ノウハウ公開」にも掲載





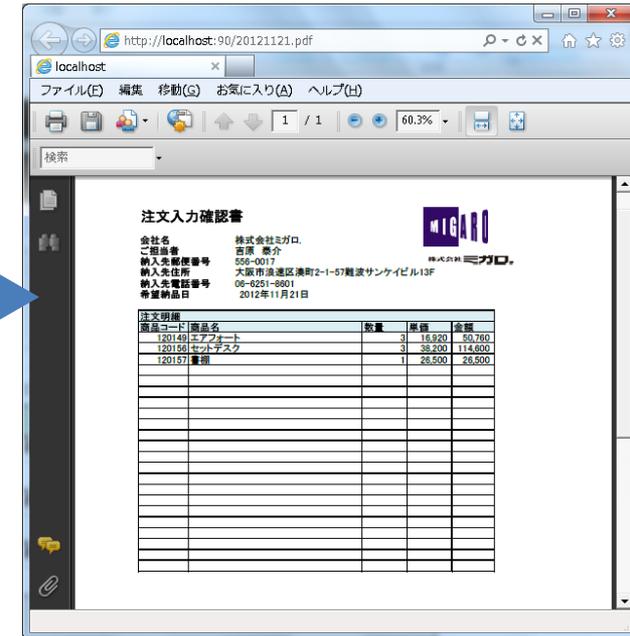
# ■ お持ち帰り資料 (Webアプリでの PDF出力方法)

- WebアプリケーションからPDF出力CGIの呼び出し

会社名	株式会社ミガロ	注文明細金額	¥ 191,860
ご担当者	吉原 泰介	送料	¥ 1,050
納入先郵便番号	556-0017	合計金額	¥ 192,910
納入先住所	大阪市浪速区湊町2-1-57難波サンケイビル13F		
納入先電話番号	06-6251-8601	希望納品日	2012/11/21

明細追加 注文確認 PDF出力

商品コード	商品名	数量	単価	金額	
120149	エアフォート	3	16,920	50,760	削除
120156	セットデスク	3	38,200	114,600	削除
120157	書棚	1	26,500	26,500	削除



## WebアプリケーションからのCGIの呼び出し例 (ソース)

```
procedure TPDFfrm.IWButton1Click(Sender: TObject);
begin
  WebApplication.NewWindow('http://migaro.co.jp/project1.exe/getpdf?code=1');
end;
```

CGIを呼び出してPDFを新しいブラウザで起動する。  
抽出条件等のパラメータは  
?以降のQuery部を編集。