

## 吉原 泰介

株式会社ミガロ.

RAD事業部 技術支援課 顧客サポート

# Delphi/400活用! スマートデバイスアプリケーション開発

拡張性の高い Delphi/400 ならでは、「jQueryMobile」を使用し、見栄えのよいスマート画面開発にチャレンジしよう。

- スマートデバイスでのWebアプリケーション活用
- Delphi/400のWebアプリケーション開発
- jQueryMobileを利用したスマートデバイス開発
- まとめ



略歴  
1978年03月26日生  
2001年龍谷大学法学部卒  
2005年07月株式会社ミガロ.入社  
2005年07月システム事業部配属  
2007年04月RAD事業部配属

現在の仕事内容  
Delphi/400とJC/400の製品試験、および月100件に及ぶ問い合わせやサポート、セミナー講師などを担当している。

## 1. スマートデバイスでのWebアプリケーション活用

ここ1、2年でiPhoneやiPadをはじめ、スマートデバイス端末が急速に普及している。これらスマートデバイス端末は個人の利用だけでなく、ビジネスツールとして、すでに導入・利用を始めている企業も少なくない。そうした背景もあり、今回は、Delphi/400でスマートデバイス向けのアプリケーションを開発するテクニックを紹介したいと考えた。これから紹介するテクニックを使えば、図1のような画面スタイルのスマートデバイスアプリケーションが実現できる。

### 【図1】

Delphi/400ではC/Sアプリケーションだけでなく、Webアプリケーションも開発できる。つまり、Webアプリケーションを開発すれば、スマートデバイス上のWebブラウザからIBM iのデータを扱うことができるようになる。

特に企業で使用するスマートデバイス

のシステム開発は昨今、端末にインストールするアプリケーションよりも、Webアプリケーションが主流になってきている。その理由としては大きく3つある。

### ●アプリケーションプログラムの統一

1つ目の理由は、アプリケーションプログラムを統一できる点である。

スマートデバイスとは言っても、iPhoneやAndroid等のOSやタブレットの規格の違いなど、端末はさまざまな種類が使われている。決められた端末(プラットフォーム)向けに開発されたアプリケーションは、異なる機種では動作させることができない。これはWindowsのアプリケーションが、Macintosh上で動作しないのと同様である。そのため、同じアプリケーションであっても、iPhone用、Android用など機種が異なれば、それぞれにあった複数のアプリケーション開発が必要になってくる。

しかしWebアプリケーションであれば

ば、Webブラウザ上で動作するため、Webブラウザが使える環境であれば共通して使用することができる。これが大きな強みとなる。

例えば、iPhoneであってもAndroidであっても同じWebアプリケーションを使うことができるので、シンプルに1つの開発だけで統一できる。もちろん、Webアプリケーションであれば、スマートデバイスだけでなく、WindowsやMac、LinuxといったPC上にも展開し、利用することができる。

### ●アプリケーションの配布・運用

2つ目の理由は、開発したアプリケーションの配布・運用面である。

通常、スマートデバイスで利用するアプリケーションは、iPhoneであってもAndroidであっても、開発したアプリケーションをマーケット上で公開したり、パッケージを配布したりする必要がある。しかし、企業で使用するアプリケーションをマーケット上で公開することは、難しいだろう。またiPhoneでは、

Apple のマーケット公開の審査も厳しいため、いろいろと準備が必要になってくる。

その点、Web アプリケーションであれば、Web サーバにアプリケーションを用意すれば、ブラウザで URL を指定するだけで利用できる。上記のような配布や運用に苦慮する必要もない。

### ●セキュリティ

3つ目の理由はセキュリティである。

スマートデバイスは手軽に携帯して利用できる半面、紛失などの可能性を十分に考慮する必要がある。特に、スマートデバイス上にインストールして動作するアプリケーションが IBM i のデータを利用していただければ、接続手段や接続情報を端末内に持っているということである。つまり、スマートデバイスを紛失した場合や、ウイルスなどで不正アクセスされてしまった場合には、社内 IBM i への接続情報や設定が漏洩する恐れが生じる。

この点においても、Web アプリケーションは端末上にアプリケーションの情報を持たず、あくまで Web サーバ上の URL にアクセスして利用するため、セキュリティ的にも安心して利用することが可能だ。

本稿では、こうした優れた特徴を持つスマートデバイス向けの Web アプリケーションを、Delphi/400 で開発する方法を説明する。ただし、Delphi/400 の開発手法は1つではなく、さまざまな手法で開発が可能である。今回はその一例の紹介であることをご承知いただきたい。

## 2. Delphi/400 の Web アプリケーション開発

Delphi/400 では、Web アプリケーション開発のために「VCL for the Web (旧称 IntraWeb)」と呼ばれる機能が用意されている。この VCL for the Web では、C/S アプリケーションと同じ開発手法で Web アプリケーションが開発できる。Delphi/400 開発者によっては非常に便利な機能である。

最初に、Delphi/400 の Web アプリケーション開発についての概略を説明

する。なお今回は、Delphi/400 VersionXE のバージョンを使用している。

### 【Web アプリケーションの開発手順】

- (1) プロジェクトの作成
- (2) コンポーネントの配置
- (3) プログラムの作成
- (4) コンパイルと実行

### 2-1. プロジェクトの作成

新規に Web アプリケーションのプロジェクトを作成するには、[ファイル] のメニューから [新規作成 | その他] を選ぶと、ダイアログが表示される。[Delphi/400 プロジェクト | VCL for the Web] より VCL for the Web Application Wizard を起動する。【図 2】

Wizard のダイアログが表示されるので、ここでは ISAPIExtension を選択して進める。【図 3】

これで、Web 用の新規プロジェクトが作成できた。【図 4】

### 2-2. コンポーネントの配置

Delphi/400 では、Web アプリケーション用に、C/S アプリケーションと類似のコンポーネントが用意されているので、同じ感覚でプログラムを作成することができる。

例えば、TIWButton、TIWRadioGroup、TIWDBGrid で画面設計をして、簡単な照会画面を作成してみる。【図 5】

### 2-3. プログラムの作成

IBM i のデータは、C/S アプリケーションと同じコンポーネントで扱うことができる。また今回は、dbExpress 接続を使用する。データの表示は TClientDataSet コンポーネントを使い、次のように構成する。【図 6】

```
TSQLConnection  
→TSQLQuery  
→TDataSetProvider  
→TClientDataSet  
→TDataSource  
→TIWDBGrid (画面表示)
```

イベントプログラムはボタンを押下時 (OnClick) に、TIWRadioGroup の指定に従って、TClientDataSet を Open するロジックとする。【ソース 1】

### 2-4. コンパイルと実行

プログラムのコンパイルを行うと、dll のアプリケーションが作成される。Windows サーバ上で稼働させた IIS (Internet Information Service) に作成した dll を配置して、Web ブラウザから実行してみよう。

PC の Web ブラウザはもちろん、スマートフォンなどの Web ブラウザからもアプリケーションを実行して、IBM i のデータを活用することができるのである。今回はスマートデバイスに画面サイズをあわせているため、PC ブラウザでは若干画面サイズを小さくしている。【図 7】

ここまで、Web アプリケーションの開発手順の概略を説明した。

なお、開発の詳しい手順については、『ミガロ. テクニカルレポート No.2 2009 年秋』掲載のレポート「Delphi/400 を利用したはじめての Web 開発」でわかりやすく解説しているのので、参考にさせていただきたい。

## 3. jQueryMobile を利用したスマートデバイス開発

さて、前章で Delphi/400 での Web アプリケーション開発手順を紹介したが、スマートデバイスで使うには、画面デザインが少々物足りないと感じる方もいるだろう。Web アプリケーションの見た目をよくするためには、画面用にデザイン画像を用意したり、JavaScript や meta タグ等を組み込んだりすることで工夫できる。

今回は、Web アプリケーションをスマートデバイスで見栄えよく完成させるために、「jQueryMobile」というオープンソースを活用する手法を紹介する。

### 3-1. jQueryMobile とは?

まず、jQueryMobile について、説明したい。【図 8】

図1



図2

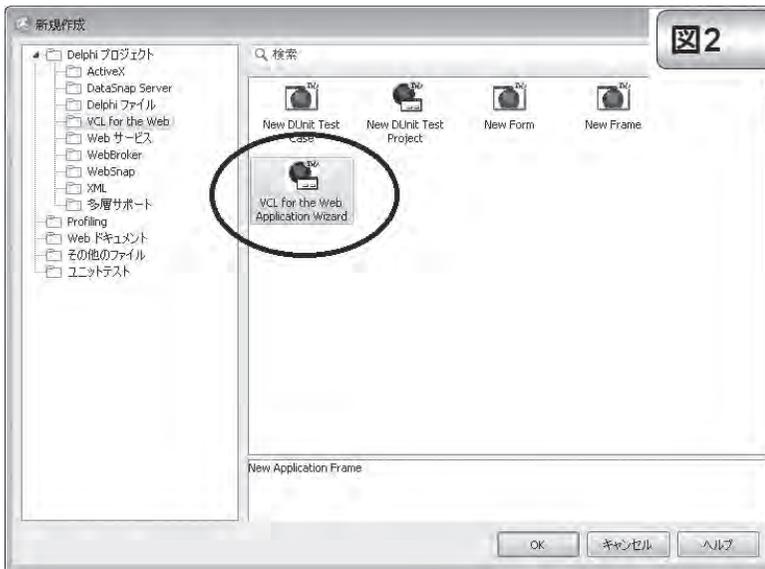


図3



jQueryMobile 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ユーザーインターフェースの画面デザインや部品を提供してくれるフレームワークである。jQueryMobile のオリジナルは海外サイトで公開されている。もちろん、日本語に訳されたリファレンスのサイトもあるので安心していただきたい。【図 9】

【jQueryMobile】

<http://jquerymobile.com>

【jQuery Mobile 1.1.0 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/)

使い方は非常に簡単で、公開されている html ソースを組み込むだけで利用することもできる。実は、Delphi/400 の「VCL for the Web」には html を組み込む機能が用意されているので、この jQueryMobile をそのまま利用することができるのである。

### 3-2. htmlの組み込み機能

早速 jQueryMobile を Delphi/400 で利用するために、html を組み込む方法を説明しよう。

まず、使用するのは TIWTemplateProcessorHTML コンポーネントである。このコンポーネントは、html を指定して、画面デザインを置き換えることができる。コンポーネントの使い方は次のとおりである。

① Web アプリケーションを配置するフォルダ内に templates というフォルダを新規作成して、利用したい画面 html を保存する。注意点としては、html ファイルの文字コードは必ず UTF-8 で保存する必要がある。【図 10】

② TIWTemplateProcessorHTML コンポーネントの Templates プロパティの Default に、①で保存した画面 html 名を設定する。【図 11】

③ Web アプリケーションの画面フォームに LayoutMgr プロパティがあるので、そ

こに TIWTemplateProcessorHTML コンポーネントを設定する。【図 12】

これだけの簡単な作業で、Delphi/400 アプリケーションの画面に html を利用することができるようになる。

### 3-3. 画面レイアウトの作成

それでは、jQueryMobile を利用して画面デザインを作成してみよう。jQueryMobile のサイト上に「単一ページテンプレート」が用意されているので、これを利用する。

まず、コピーした html を templates フォルダに保存する。【ソース 2】

【参考 URL】

[http://dev.screw-axis.com/doc/jquery\\_mobile/#/doc/jquery\\_mobile/components/pages\\_dialogs/single\\_page\\_template/](http://dev.screw-axis.com/doc/jquery_mobile/#/doc/jquery_mobile/components/pages_dialogs/single_page_template/)

続いて、TIWTemplateProcessorHTML コンポーネントに html を設定して、実際にアプリケーションを実行してみよう。実行すると、今まで作成していた画面（図 7）とは、全く異なる html が表示されることが確認できるだろう。【図 13】

ところで、html をうまく組み込むことができたが、画面は jQueryMobile サイトからコピーしたテンプレートの内容のままだ。そこで、html のページタイトル、コンテンツ、フッターを書き換えてみよう。【ソース 3】

再度実行してアプリケーション画面を確認すると、書き換えた html が反映されている。【図 14】

ちなみに、html は Delphi/400 の開発環境で開くこともできる。つまり、Delphi/400 ソースと同じように、html を編集することができる。【図 15】

以上で、スマートデバイスらしい画面デザインを取り込むことができた。さらに見た目だけでなく実は、この画面はスマートデバイスの端末の形態にあわせて、画面を自動調整してくれるのである。もちろん縦、横の画面回転にも対応している。

### 3-4. コンポーネントとhtmlの連携

html で、スマートデバイス用の画面デ

ザインを適用することができた。ただし、今の状態では、Delphi/400 で作成したコンポーネントが表示されていない。ここからは、コンポーネントを html と連携させる方法を説明していく。

連携させる方法は簡単である。html に、コンポーネントの名前を次のように追記すればよい。

```
%コンポーネント名%
```

では、実際に、html のコンテンツ部分にコンポーネント名を組み込んでみよう。

コンパイルして実行すると、html の画面の中にコンポーネントが組み込まれ、表示されていることが確認できる。つまり、画面だけ html で表示して、裏では Delphi/400 のプログラムがそのまま動作しているという仕組みである。【ソース 4】【図 16】【図 17】

### 3-5. ラジオグループのカスタマイズ

さて、コンポーネントが表示されたので、ここからはコンポーネントの見たい目についても、jQueryMobile を活用してカスタマイズしていく。

まず TIWRadioGroup については、jQueryMobile のサイト上にラジオボタンのサンプルが用意されている。このソースをコピーして、部品の名前を TIWRadioGroup のコンポーネント名 (Name プロパティ値) に変更し、html を書き換えよう。【ソース 5】

【参考 URL】

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

アプリケーションを実行して画面を表示すると、ラジオグループがカスタマイズされていることが確認できる。【図 18】

### 3-6. グリッドのカスタマイズ

次に、データを表示しているグリッドをカスタマイズする。グリッドについては、jQueryMobile のサイト上にリストビューとしてサンプルが用意されている。

図4

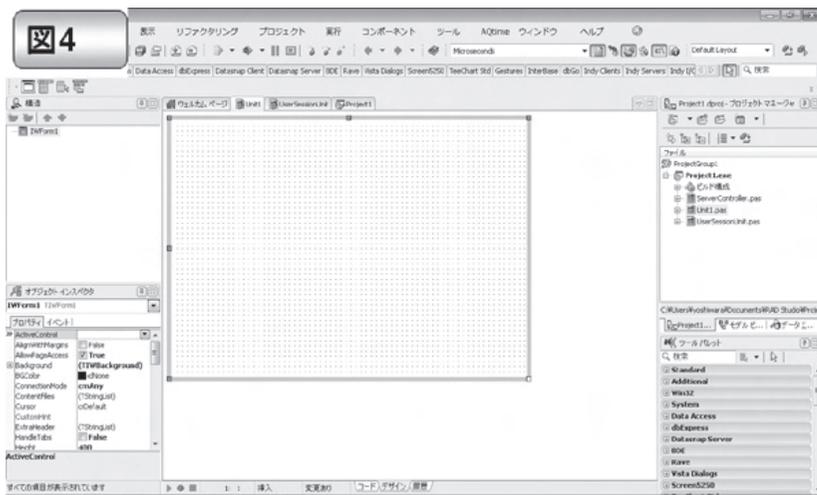
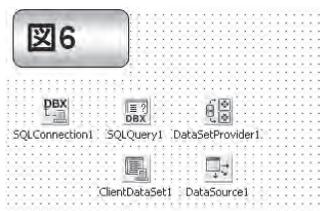


図5



図6



ソース1

ソース1: 表示ボタン押下時 Delphi/400

```

procedure TForm1.IWButton1Click(Sender: TObject);
begin
  //データを閉じる
  ClientDataSet1.Close;
  //SQLのバインド変数に画面の指定をセット (ここでは区分)
  SQLQuery1.ParamByName('区分').AsInteger := IWRadioGroup1.ItemIndex;
  //データを表示
  ClientDataSet1.Open;
end;
  
```

#### 【参考 URL】

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

このhtmlを参考にして、TIWDBGridをカスタマイズできる。

ただし、データの量によっては、可変で変更されるリストビューをhtmlで事前に定義することが難しい。そのため、TIWDBGridに関しては、Delphi/400のプログラムの中で動的にhtmlを作成することにする。

#### ●動的なリストビュー

まず、ClientDataSetのデータ量によって、リストビューのhtmlソースを動的に作成する関数を用意する。【ソース6】

そして、html上で`{% IWDBGrid1 %}`と記述している部分を、`{% ListView %}`へ変更する。これによって、TIWDBGridを表示せずに、新しくListViewの描画が行えるようになる。このリストビューは、TIWTemplateProcessorHTMLコンポーネントのUnknownTagイベントで、関数を呼び出して作成している。【ソース7】

アプリケーションを実行して、表示ボタンをタッチしてみよう。すると、データの表示がグリッドではなく、リストビュー形式で表示される。【図19】

ここまでで、元の画面部品を全てスマートデバイス向けにカスタマイズすることができた。

最後におまけとして、リストビューに画像表示も組み込んでみよう。

フィールドの値と一致する画像ファイルがWebサーバに保存されていることが前提とするが、`<img>`タグのhtmlコードを追加すれば、簡単にリストビュー上に画像を組み込むことができる。【ソース8】これで画像付きの照会アプリケーションの完成である。【図20】

## 4.まとめ

さて本稿では、Delphi/400がhtmlを組み込めるという機能を着眼点として、スマートデバイス向けのWebアプリケーション開発方法を説明してきた。拡張性が高いDelphi/400ならではのスマートデバイス対応である。

今回紹介したjQueryMobileをうまく活用すれば、誰でも容易に高機能なスマートデバイス画面を開発できることをお伝えできただろうか。もちろんhtmlを組み込まなくとも、Webアプリケーションを開発することはできるので、あくまでスマートデバイスに簡単に対応するための1つの手法と考えていただきたい。

またDelphi/400では、jQueryMobileに限らず、公開されているさまざまなhtmlやJavaScriptの技術を利用することが可能だ。ぜひWebアプリケーション開発の際には、見栄えのよいスマートデバイス画面にチャレンジしてほしい。その際、本稿のテクニックが参考になれば幸いである。

M

図7



図8

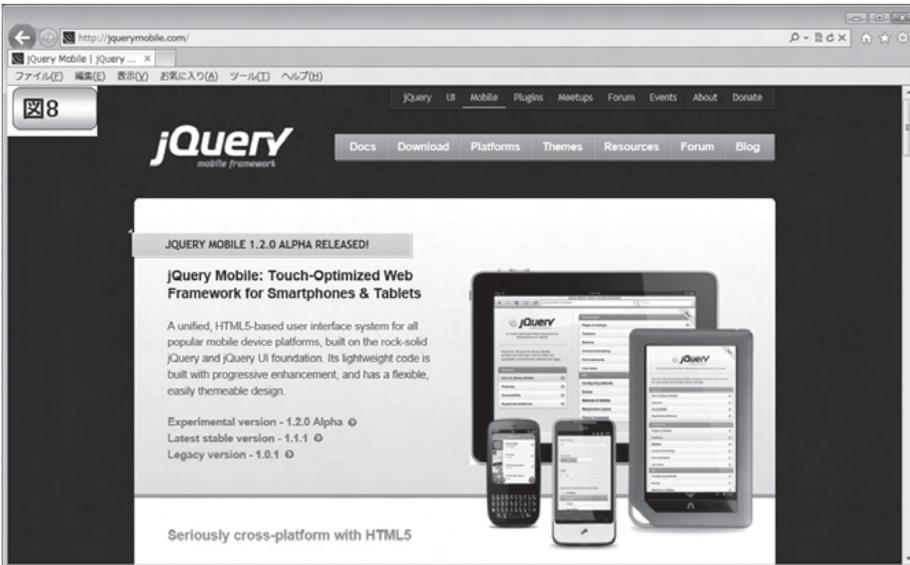


図9

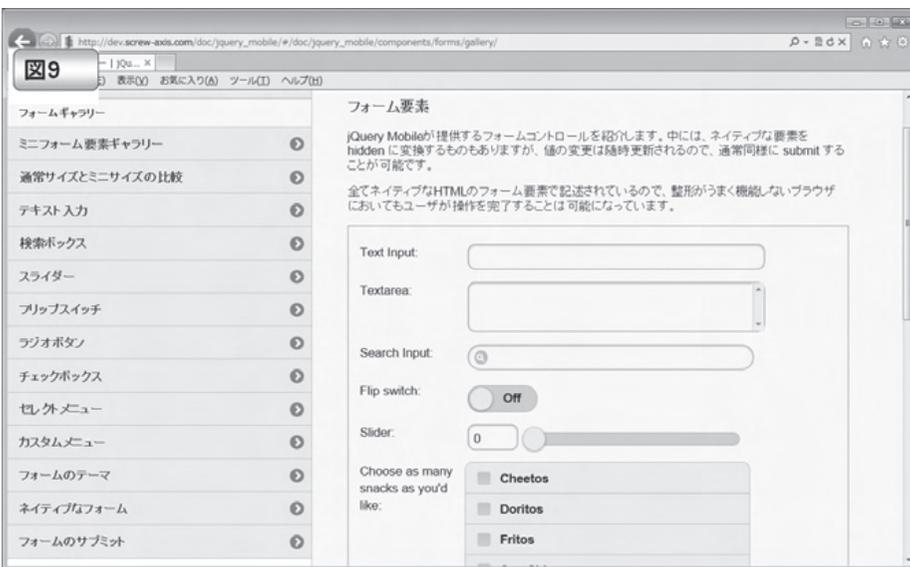


図10



図11

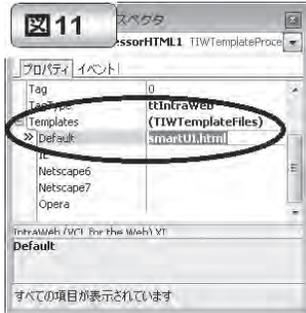


図12



ソース2

```

ソース2: 単一ページテンプレートhtml      html
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.css" />
    <script type="text/javascript" src="http://code.jquery.com/jquery-1.7.1.min.js"></script>
    <script type="text/javascript" src="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.js"></script>
  </head>
  <body>

  <div data-role="page">
    <div data-role="header">
      <h1>Page Title</h1>
    </div><!-- /header -->
    <div data-role="content">
      <p>Page content goes here.</p>
    </div><!-- /content -->
    <div data-role="footer">
      <h4>Page Footer</h4>
    </div><!-- /footer -->
  </div><!-- /page -->
</body>
</html>

```

図13



### ソース3

```
ソース3: htmlの変更部分      html
<div data-role="page">
  <div data-role="header">
    <h1>製品一覧照会</h1>
  </div><!-- /header -->

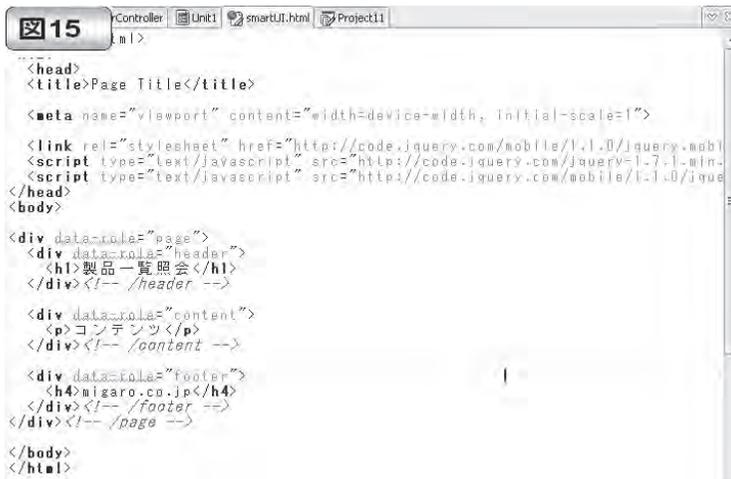
  <div data-role="content">
    <p>コンテンツ</p>
  </div><!-- /content -->

  <div data-role="footer">
    <h4>migaro.co.jp</h4>
  </div><!-- /footer -->
</div><!-- /page -->
```

図14

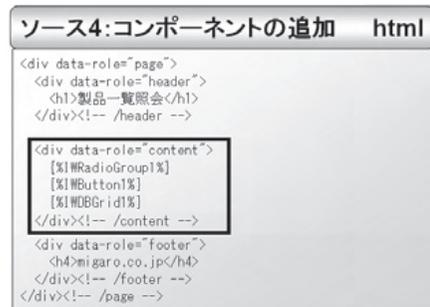


図15



```
<head>
<title>Page Title</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.css">
<script type="text/javascript" src="http://code.jquery.com/jquery-1.7.1.min.js"></script>
<script type="text/javascript" src="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.js"></script>
</head>
<body>
<div data-role="page">
<div data-role="header">
<h1>製品一覧照会</h1>
</div><!-- /header -->
<div data-role="content">
<p>コンテンツ</p>
</div><!-- /content -->
<div data-role="footer">
<h4>migaro.co.jp</h4>
</div><!-- /footer -->
</div><!-- /page -->
</body>
</html>
```

## ソース4

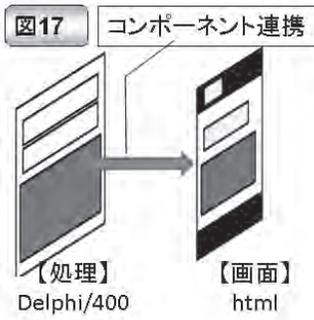


```
ソース4:コンポーネントの追加 html
<div data-role="page">
<div data-role="header">
<h1>製品一覧照会</h1>
</div><!-- /header -->
<div data-role="content">
[%#RadioGroup1%]
[%#Button1%]
[%#MDBGrid1%]
</div><!-- /content -->
<div data-role="footer">
<h4>migaro.co.jp</h4>
</div><!-- /footer -->
</div><!-- /page -->
```

図16



図17



ソース5

ソース5:ラジオグループの書き換え html

```

<fieldset data-role="controlgroup" data-type="horizontal" data-role="fieldcontain">
  <legend>ディスプレイの区分を選択してください。</legend>
  <div type="RADIOGROUP" class="IWRADIOGROUP1CSS" style="z-index: 100; width: 300px; height: 36px;
    font-weight:normal;font-style:normal;text-decoration: none;font-size:13px;" id="IWRADIOGROUP1"
    name="IWRADIOGROUP1" tabindex="3">
    <input type="radio" name="IWRADIOGROUP1_INPUT" id="IWRADIOGROUP1_INPUT_1" value="0" checked="true" />
    <label for="IWRADIOGROUP1_INPUT_1">液晶</label>
    <input type="radio" name="IWRADIOGROUP1_INPUT" id="IWRADIOGROUP1_INPUT_2" value="1" />
    <label for="IWRADIOGROUP1_INPUT_2">プラズマ</label>
    <input type="radio" name="IWRADIOGROUP1_INPUT" id="IWRADIOGROUP1_INPUT_3" value="2" />
    <label for="IWRADIOGROUP1_INPUT_3">その他</label>
  </div>
</fieldset>
  
```

図18



## ソース6

```
ソース6 リストビューの動的作成関数 Delphi/400

//宣言
public
function RenderI#Grid: String;
end;

//実装部
function TForm1.RenderI#Grid: String;
var
ul : TStringList;
begin
begin
//グリッドのhtml文を動的に作成する。
ul := TStringList.Create;
ul.Add('<ul data-role="listview">');
//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('<h3>' + ClientDataSet1.FieldByName('製品コード').AsString);
ul.Add(ClientDataSet1.FieldByName('製品名').AsString + '</h3>');
ul.Add('<p>価格:' + ClientDataSet1.FieldByName('価格').AsString + '</p>');
ul.Add('<p>在庫:' + ClientDataSet1.FieldByName('在庫数').AsString + '</p>');
ul.Add('</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;
end;
```

メインデータ<h3>
サブデータ<p>

## ソース7

```
ソース7:リストビューの動的作成 Delphi/400

procedure TForm1.ITemplateProcessorHTMLUnknownTag(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 := RenderI#Grid;
end;
end;
```

図19



## ソース8

```
ソース8: 画像データの組み込み Delphi/400

function TForm1.RenderI#Grid: String;
var
  ul : TStringList;
begin
  ~~~~

  while not(ClientDataSet1.Eof) do
  begin
    ul.Add('<li>');
    ul.Add('');
    ul.Add('<h3>' + ClientDataSet1.FieldName('製品コード').AsString);
    ul.Add(ClientDataSet1.FieldName('製品名').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('</li>');
    ClientDataSet1.Next;
  end;
  ul.Add('</ul>');
  ~~~~

```

画像<img>

図20

