

福岡 浩行

株式会社ミガロ.

システム事業部 システム2課

Delphi/400を利用したはじめてのWeb開発 —VCL for the Webによる簡単な開発方法!

開発者であれば、Web アプリケーションを開発してみたいと思うだろう。

ここでは

Delphi/400 の VCL for the Web で可能になった Web アプリケーション開発方法を紹介します。



略歴

1984年11月11日生れ
2007年関西学院大学理工学部卒
2007年04月株式会社ミガロ、入社
2007年04月システム事業部配属

現在の仕事内容

Delphi/400 や RPG の開発業務を担当。Web 開発技術に関する知識 (JavaScript、CGI、PHP 等) を習得し、オールマイティな提案や開発が担える SE を目指している。

- はじめに
- VCL for the Webでの開発用途/形態
- VCL for the Webでの画面設計
- VCL for the Webでのプログラムロジック開発
- VCL for the Webでの応用開発
- 開発端末でのWebサーバー環境構築
- おわりに

はじめに

近年、ユビキタス社会が確立されつつある中、開発者であれば一度は Web アプリケーションを開発してみたいと思ったことはないだろうか。私もその中の一人で、企業にある蓄積された情報を Web 化することで、より情報を有効に扱えるものと考えている。

しかし、簡単に Web アプリケーションを作ることにちょっと考えてみても、Java や PHP 等の新しい技術を習得する必要があったり、既存のシステムとの調整も必要になったりする。さらに、これらのさまざまな不安要素を解決するには大幅な時間を費やすことが求められる。

Delphi/400 の Web 開発機能である「VCL for the Web (旧 IntraWeb)」では、従来のネイティブ Windows アプリケーション開発と同じような手法で、Web アプリケーション開発を行うことができる。

本稿では、Delphi/400 の VCL for

the Web を利用した基本的な開発方法と各種テクニックの紹介を目的とする。

そこで今回は、簡単な商品検索の Web 照会アプリケーション作成を例に挙げて、VCL for the Web の基本的な手順を、以下の流れにそって説明していく。(Web 照会アプリケーションの完成画面は、手順最後の図5を参照)【図5】

開発形態の決定→画面の設計→プログラムロジック開発

そして、応用テクニックと開発端末での Web サーバー検証環境の作成についても簡単にふれていきたい。

VCL for the Web での開発用途 / 形態

アプリケーションの種類

VCL for the Web (以下 IntraWeb) で Web アプリケーションを動作させるには、以下の3種類の方法がある。

A スタンドアローンモード

プログラミング作業と実行を手軽に行う場合、このモードを選択する。このモードでは Web サーバーも IntraWeb が提供するため、Windows アプリケーションと同じ実行形式が作られる。

B アプリケーションモード

既存の Web サーバー (IIS) を利用し、簡単にプログラミングを行いたい場合はこのモードを選択する。メモリや DB コネクションに余裕がある場合は、このモードが一番利用しやすい。

C ページモード

Web アプリケーションに WebSnap または WebBroker を利用する。リソースを再利用して大量の要求を受け付ける場合は、このモードを利用する。

用途によって、さまざまなモードを選択し、開発を行っていく必要がある。

本稿では初心者の方でも一番手軽に行うことができる、アプリケーションモードを利用した開発方法を記述する。

アプリケーションモードのWeb開発

それでは、アプリケーションモードを用いての Web プログラミングの開発方法を順に説明していく。

まず、新規プロジェクトを作成する。この時、新規プロジェクトを作成するには [ファイル] → [その他] → [Delphi プロジェクト / VCL for the Web] → [VCL for the Web Application Wizard] を選択する。【図 1a】 【図 1b】

すると、Web アプリケーションのウィザードが開始される。ここで表示される Application Type というのが、上記で説明した 3 種のモードに該当する。ここでは、アプリケーションモードの ISAPI Extension を選択する。

Option は特に個別に設定する必要がないため、初期値の状態を設定し、Project Name、Project Directory は任意の値を設定する。すべての設定が完了した後に、[OK] ボタンを押下すれば、アプリケーションモードでの開発画面に遷移する。【図 2】

開発画面には、ServerController.pas、Unit.pas、UserSessionUnit.pas の 3 種が自動生成される。ServerController はその名の通り、サーバーのコントロールを司るファイルで、ブラウザの戻るボタンや IntraWeb 終了時の制御を行う等のことができる。Unit ファイルは実際に開発を行う画面で、このファイルにコンポーネントを貼り付けて開発を行う。UserSessionUnit は、アプリケーション開発を行う時の DataModule と同じ扱いになる。

今回は、条件を指定して検索ボタンを押下すると商品の検索を行うことができる、という簡単な照会システムの作成手順を説明する。

VCL for the Web での画面設計

IntraWeb で使用できるコンポーネントは、C/S アプリケーションを開発するコンポーネントとは違って来る。ツールパレットの中で、頭に IW がつくコンポーネント群が使用できるコンポーネントになる。簡単な開発であれば IW Standard 群が主に使用される。

今回の開発で使用するコンポーネン

トとしては、TIWEdit、TIWGrid、TIWButton、TIWImageFile、TIWLabel、TIWRegion、TIWLink、TIWRadioButton がある。コンポーネントの配置方法は従来のアプリケーション開発と変わらず、表示を行う箇所にコンポーネントの配置を行い、必要に応じてプロパティの設定を行う。【図 3】

簡単に各コンポーネントと主なプロパティの説明を、以下に記述する。

● TIWEdit

ブラウザ画面で文字等の入力制御を行う。プロパティの Maxlength に数値を入れることで、最大入力文字数が指定できる。

● TIWGrid

表形式の画面を出力する。HTML と言えば TABLE に相当する。プロパティの UseFrame を True にすることで、開発画面上で指定した幅を固定にできる。False の場合は、出力するデータによって Grid の幅が変化する。また、Line の設定や列の間隔を設定することも可能。

● TIWButton

画面上でクリックすることで、イベントを呼び出す。

● TIWImageFile

画像を画面上に表示するコンポーネント。プロパティの ImageFile の File Name または画像保管場所の URL を指定することで、画像を表示できる。

● TIWLabel

テキストを表示するコンポーネント。

● TIWRegion

TPanel と同じ働きをするコンポーネント。

● TIWLink

IntraWeb 内の画面に遷移するコンポーネント。似たコンポーネントに TIWURL があり、こちらのコンポーネントは IntraWeb 外の画面に遷移する時に使用する。

● TIWRadioButton

複数の選択肢から条件を指定できるコンポーネント。プロパティの Group で同じ名前を指定したものが同一選択対象になる。

画面の構成が決定した後に、IBM i との連携を行うための非ビジュアルコン

ポーネントの配置を行う。よく Web アプリケーションで課題となるのが、サーバーからの応答時間の問題である。データベースとの接続方法はさまざまな方法があるが、Delphi/400 で最もレスポンスが早い単一方向データセットを採用している DBExpress を使用することが多い。

そのため、今回の説明も DBExpress 接続方式を採用する。DBExpress の TSQLConnection、TSQLQuery を UserSessionUnit のフォームに配置し、TSQLConnection を右クリックして接続設定を図のようにする。【図 4】

ここで設定する DataBaseName と HostName には、Configuration で扱う接続名を設定する。これらの設定はプログラム内部で指定することが可能だが、必要に応じてプロパティで設定をする。

VCL for the Web でのプログラムロジック開発

次にプログラム内部の設計に入る。プログラム内部の設計は従来のアプリケーション開発と変わらず、各コンポーネントにあるイベントにプログラムを記述していけばよい。

例えば、メッセージのみの出力だけを考えてみると、Windows アプリケーションでは ShowMessage 関数があるが、同様に Web アプリケーションでも ShowMessage 関数が存在する。

<Windows アプリケーション>

```
Application.ShowMessage ('Hello World !');
```

<Web アプリケーション>

```
WebApplication.ShowMessage ('Hello World !');
```

OnClick イベント

今回のプログラムでは、TIWButton にある OnClick イベントに検索を実行するプログラムを記述する。ボタンを押下することで、Query を利用して IBM i に登録されているデータベースにアクセスを行い、検索条件に一致したレコードを取得する。【ソース 1】

お気づきかもしれないが、OnClick

イベントは通常の OnClick イベントと OnAsyncClick イベントの 2 種類がある。これら 2 つのイベントはボタンを押下時に呼び出されるイベントである。

まず OnClick イベントでは、押下時にデータをサーバーに通知して、処理結果を得た後にページ全体をロードする (Submit を行う) というイベントになっている。

一方、OnAsyncClick イベントは、指定した URL から XML ドキュメントを読み込む機能を使い、ユーザーの操作や画面描画等と並行する Ajax のようなイベントになる。

そのため、画面上のみで制御を行う場合は OnAsyncClick にイベントを記述するほうがよいが、逆のパターンでは OnClick イベントに記述するほうがよい。また、OnAsyncClick イベントでは、Submit が行われなためコンポーネントの画面表示切替処理などの制御が行えない制約がつく。

明細部への画像表示

また、今回は明細部の中に画像を表示する仕様にする。画像ファイルはサーバー側の特定のフォルダに保存されていることを前提に、IBM i のデータベースにはそのパスを保持するようにする。

画像を特定のパスで読み込むためには、前述した TIWImageFile を使用する。このコンポーネントをプログラム内部で生成するようにし、URL プロパティに画像を保存しているパスを指定することで、明細部の中に画像の表示を行うことができる。

DLLの作成

一通りの制御開発が終了した時点で、[プロジェクト] → [再構築] を選択し、プログラムの DLL を作成する。[プロジェクト] → [コンパイル] を選択しても問題はないが、再構築を行うことで中間ファイル等を再生成してくれるため、再構築を勧める。

この作成した DLL を外部からアクセスできる Web サーバーに設置し (それに関連する画像とのファイルも設置する)、ブラウザの URL に DLL が保管されているパスを指定する。

すると、Web アプリケーションが起動する。【図 5】

VCL for the Web での応用開発

さらに、一歩進んだことを行ってみよう。よく他のサイトをみるとクリックした画像が大きく表示されたり、文字のフォントを統一していたりするサイトが多く目立つ。IntraWeb でも JavaScript やスタイルシートが利用できる。

今回の開発では、明細に表示した画像をクリックすることで新しい画面を作り、画像を表示する JavaScript と、明細に表示する画像の大きさを統一するスタイルシートを組み込むことにする。

スタイルシートとの連携

スタイルシートとは Web ページのレイアウトを定義する技術であり、一般的にスタイルシートと言われると、スタイルシート言語の 1 つである CSS を指す。本文でもスタイルシートのことを CSS として明記する。

HTML を少しでも知っている方は、CSS を読み込む際に、その記述を HTML のヘッダにすることをご存知だろう。IntraWeb では、Form のプロパティに ExtraHeader というのがある。ここに CSS を読み込む処理を記述するか、プログラム内部で記述するかの 2 種類の方法がある。今回の開発では、動的に使用できるようにプログラム内部に記述する。【ソース 2】

次に、CSS を利用するコンポーネントに設定を行う。CSS の設定できるコンポーネントには、プロパティに CSS というのがある。ここに、CSS に記述されているセクタを記述すれば、そのコンポーネントはスタイルシートに従った表記になる。【ソース 3】

JavaScriptとの連携

JavaScript とは Web ページに動きを追加する簡易言語のことで、画像をクリックした時に新しいページを表示させたり、TAB の移動といった制御等を行うことができる。

通常の Web 開発では、HEAD タグ内に JavaScript を記述する。IntraWeb でも同様に HEAD タグ内に JavaScript の記述が行えるが、今回は明細の画像をクリックすることで詳細な画像を表示させる仕様にする事から、コンポーネン

トに JavaScript を記述する方針をとる。

コンポーネントに JavaScript を記述するには、プロパティの ScriptEvents に記述するか、プログラム内部で記述するかの 2 種類がある。静的なコンポーネントであれば ScriptEvents に記述するほうがよいが、今回は動的に画像ファイルを作成しているため、プログラム内部で JavaScript を記述する。【ソース 3】

ソース 3 のように記述を行えば、実行し、画像をクリックした時に仕様どおりの動きを行うことができる。他にも JavaScript を駆使すれば、Enter キーでフォーカスの移動が行えたり、コンポーネントのイベントと連携を行ったりすることが可能になる。

開発端末での Web サーバー環境構築

ここまでは Web アプリケーション作成を説明してきたが、作成したアプリケーションを Web サーバー上で動作検証するためのテクニックを説明する。

ここでは開発端末に Web サーバーを構築し、作成した DLL が正常に動作するか検証を行えるようにする。

今回の動作環境は、Windows XP の IIS (バージョン 5.1) を使用する。Windows のコントロールパネルの中に [管理ツール] → [インターネットインフォメーションサービス] (以下 IIS) というのがある。(ない場合は [プログラムの追加と削除] → [Windows コンポーネントの追加と削除] からインストールが可能)。この IIS は、Microsoft 社のインターネットサーバーソフトウェアで、この IIS を構築することで Web サーバーを作ることができる。

構築するやり方として、IIS を起動すると、Web サイトという項目の下に既存の Web サイトという項目がある。この項目に指定したフォルダが Web サーバーの仮想ディレクトリとなり、外部からのアクセスが可能となる。既存の Web サイトの項目で、右クリックを行うと [新規作成] という項目がある。この項目を選択し、ウィザードに従って仮想ディレクトリを作成する。

ここで注意しなければならないのが、DLL を実行するためのアクセス許可であり、ISAPI アプリケーションや CGI

図3

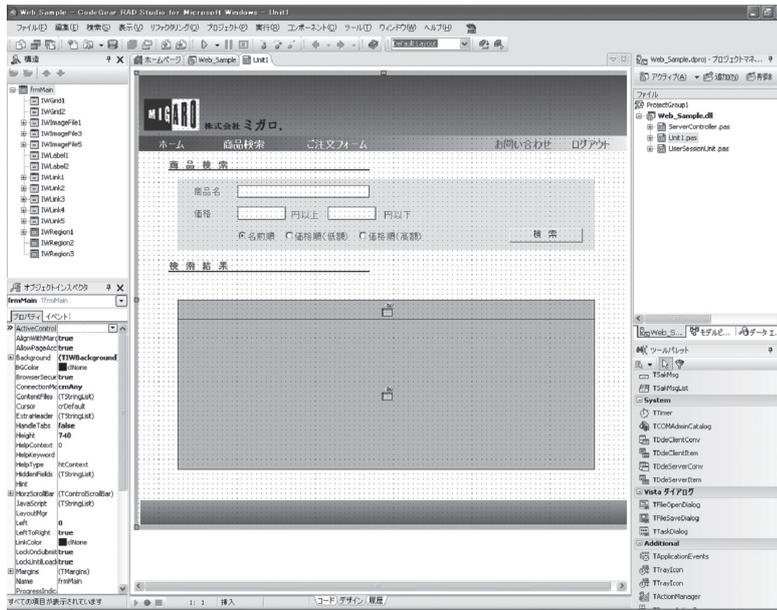


図4

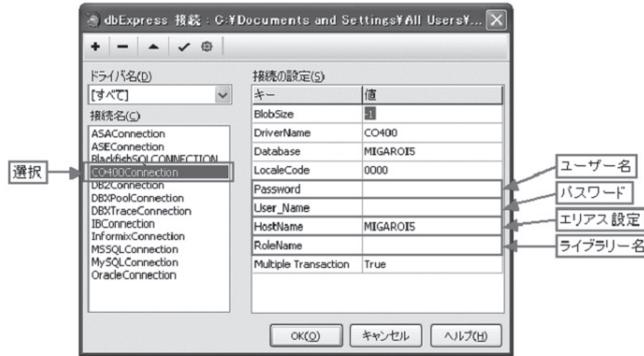


図5



などを実行するための許可を追加しなければならない。【図 6】

作られたディレクトリに DLL 等のファイルを設置すれば、ブラウザからアクセスが可能となる。

ブラウザからアクセスする際は、以下のようにアクセスを行う。

http:// サーバー名 (コンピュータ名) /
設定した仮想ディレクトリ / DLL ファイル

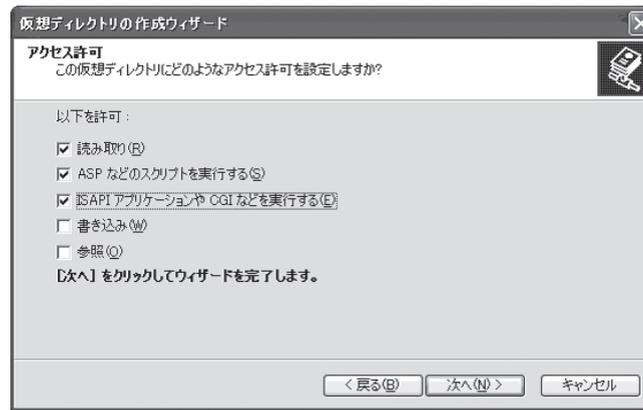
おわりに

はじめに記述したように、私の中では Delphi の C/S アプリケーション開発と同様に Web アプリケーションが作成できることがおおいに嬉しかった。これまでの IntraWeb は、主に社内ネットワークアプリケーションとして使用されてきたが、機能が追加されバージョンが上がったことで、Web アプリケーションとしての活用が見込める存在となった。

今回の事例は照会のみアプリケーションであったが、登録やメール送信、CGI などもこのアプリケーションに組み込むことが可能である。今後の Web 開発にぜひとも役立てていただきたい。

■

図6



ソース1

```

Unit1.pas - TeraPad
ファイル(F) 編集(E) 検索(S) 表示(V) ウィンドウ(W) ツール(T) ヘルプ(H)
10, 20, 30, 40, 50, 60, 70, 80
55
56 {*****}
57 □目的: 「検索」 ボタン押下時処理
58 □引数: なし
59 戻値: なし
60 {*****}
61 procedure TFormMain.btnSearchClick(Sender: TObject);
62 const
63 //SQLメイン
64 sSQLMain = 'SELECT * FROM SHMAS WHERE SHDLFG IS NOT NULL ';
65
66 //SQL条件
67 sSQLStr1 = 'AND SHNAME LIKE :SHNAME '; //商品名
68 sSQLStr2 = 'AND SHKKKU >= :SHKKKU1 '; //価格
69 sSQLStr3 = 'AND SHKKKU <= :SHKKKU2 ';
70
71 //並び替え
72 sSQLOrder1 = 'ORDER BY SHNAME ASC';
73 sSQLOrder2 = 'ORDER BY SHKKKU ASC, SHNAME ASC';
74 sSQLOrder3 = 'ORDER BY SHKKKU DESC, SHNAME ASC';
75 begin
76
77 with UserSession do
78 begin
79 if qryMain.Active then qryMain.Close;
80 qryMain.SQL.Clear;
81 qryMain.SQL.Add(sSQLMain);
82
83 //条件
84 //商品名
85 if Trim(edtSHNAME.Text) <> '' then
86 begin
87 qryMain.SQL.Add(sSQLStr1);
88 qryMain.ParamByName('SHNAME').AsString := '%' + edtSHNAME.Text + '%';
89 end;
90 //価格 (最低)
91 if Trim(edtSHKKKUF.Text) <> '' then
92 begin
93 qryMain.SQL.Add(sSQLStr2);
94 qryMain.ParamByName('SHKKKU1').AsInteger := StrToInt(edtSHKKKUF.Text);
95 end;
96 //価格 (最高)
97 if Trim(edtSHKKKUT.Text) <> '' then
98 begin
99 qryMain.SQL.Add(sSQLStr3);
100 qryMain.ParamByName('SHKKKU2').AsInteger := StrToInt(edtSHKKKUT.Text);
101 end;
102
103 //並び替え
104 if rbtnG1.Checked then qryMain.SQL.Add(sSQLOrder1)
105 else if rbtnG2.Checked then qryMain.SQL.Add(sSQLOrder2)
106 else qryMain.SQL.Add(sSQLOrder3);
107
108 //SQL実行
109 try
110 qryMain.Open;
111
112 //Gridに描画する
113 GridDraw;
114 finally
115 if qryMain.Active then qryMain.Close;
116 end;
117 end;
118 end;
119

```

```
Unit1.pas - TeraPad
ファイル(F) 編集(E) 検索(S) 表示(V) ウィンドウ(W) ツール(T) ヘルプ(H)
|0, |10, |20, |30, |40, |50, |60, |70, |80,
212 {*****}
213 □目的: 画面読み込み時
214 □引数: なし
215 □戻値: なし
216 *****}
217
218 procedure TfrmMain.IWebFormRender(Sender: TObject);
219 const
220   Row0 = '300';
221   Row1 = '100';
222   Row2 = '50';
223 begin
224   //ImageファイルHOST取得
225   imgHostPath := 'http://' + WebApplication.Request.Host + '/MTR_Sample/Files/image';
226   //CSSファイルHOST取得
227   cssHostPath := 'http://' + WebApplication.Request.Host + '/MTR_Sample/Files/css';
228   //CSS読み込み
229   ExtraHeader.Add(
230     '<link rel="stylesheet" type="text/css" href="'+cssHostPath+'/System.css" />');
231
232   //タイトル部
233   wgTitle.Cell[0,0].Width := Row0;
234   wgTitle.Cell[0,1].Width := Row1;
235   wgTitle.Cell[0,2].Width := Row2;
236   wgTitle.Cell[0,0].BackColor := clWebKHAKI;
237   wgTitle.Cell[0,1].BackColor := clWebKHAKI;
238   wgTitle.Cell[0,2].BackColor := clWebKHAKI;
239   wgTitle.Cell[0,3].BackColor := clWebKHAKI;
240   wgTitle.Cell[0,0].Alignment := taLeftJustify;
241   wgTitle.Cell[0,1].Alignment := taRightJustify;
242   wgTitle.Cell[0,2].Alignment := taRightJustify;
243   wgTitle.Cell[0,3].Alignment := taCenter;
244   wgTitle.Cell[0,0].Text := '商品名';
245   wgTitle.Cell[0,1].Text := '価格';
246   wgTitle.Cell[0,2].Text := '在庫数';
247   wgTitle.Cell[0,3].Text := '写真';
248   //名細部
249   wgMain.Cell[0,0].Width := Row0;
250   wgMain.Cell[0,1].Width := Row1;
251   wgMain.Cell[0,2].Width := Row2;
252 end;
253
254 initialization
255   TfrmMain.SetAsMainForm;
256
257 end.
258
```

```

Unit1.pas - TeraPad
ファイル(F) 編集(E) 検索(S) 表示(V) ウィンドウ(W) ツール(T) ヘルプ(H)
119
120 {*****}
121 □目的: Grid描画処理
122 □引数: なし
123 戻値: なし
124 {*****}
125 procedure TfrmMain.GridDraw;
126 const
127   Row0 = '300';
128   Row1 = '100';
129   Row2 = '50';
130 var
131   iRow : Integer; //変数
132   str  : string;
133 begin
134   //初期化
135   wgMain.RowCount := 1;
136   iRow := 0;
137   str := '';
138
139   with UserSession.qryMain do
140   begin
141     //明細部
142     wgMain.Refresh;
143     while not(EOF) do
144     begin
145       //商品名
146       wgMain.Cell[iRow,0].Width := Row0;
147       wgMain.Cell[iRow,0].Alignment := taLeftJustify;
148       wgMain.Cell[iRow,0].Text := FieldByName('SHNAME').AsString;
149       //価格
150       wgMain.Cell[iRow,1].Width := Row1;
151       wgMain.Cell[iRow,1].Alignment := taRightJustify;
152       wgMain.Cell[iRow,1].Text := FormatFloat('#,##',FieldByName('SHKKKU').AsInteger);
153       //在庫数
154       wgMain.Cell[iRow,2].Width := Row2;
155       wgMain.Cell[iRow,2].Alignment := taRightJustify;
156       wgMain.Cell[iRow,2].Text := FormatFloat('#,##',FieldByName('SHZIKO').AsInteger);
157       //写真
158       if FieldByName('SHPHTO').AsString <> '' then
159       begin
160         with wgMain.Cell[iRow, 3] do
161         begin
162           Alignment := taCenter;
163           Control := TIWImageFile.Create(Self);
164
165           //画像ファイルを作成する
166           with TIWImageFile(Control) do
167           begin
168             ImageFile.URL := imgHostPath+'/' +FieldByName('SHPHTO').AsString;
169             StyleRenderOptions.RenderSize := False;
170             AutoSize := True;
171             Css := 'image';
172             str := imgHostPath + '/' +FieldByName('SHPHTO').AsString;
173             str := StringReplace(str,'¥','/',[rfReplaceAll]);
174
175             //JavaScriptイベント
176             ScriptEvents.Values['onClick'] :=
177             'newWin = window.open("", "new", "width=480,height=360,resizable=1,menubar=1,sta
178             'newWin.location = "' +str+'";'+
179             'newWin.focus()';
180
181           end;
182         end;
183       end;
184
185       //行追加
186       wgMain.RowCount := wgMain.RowCount+1;
187
188       //変数追加
189       Inc(iRow);
190
191       Next;
192     end;
193
194     //該当レコードが0件の処理
195     if (iRow=0) then
196     begin
197       lblComment.Caption := '該当するレコードが存在しませんでした。';
198       wgTitle.Visible := False;
199       wgMain.Visible := False;
200     end
201     else
202     begin
203       //最終行削除
204       wgMain.RowCount := wgMain.RowCount-1;
205       lblComment.Caption := IntToStr(wgMain.RowCount)+' 件ヒットしました。';
206       wgTitle.Visible := True;
207       wgMain.Visible := True;
208     end;
209     lblComment.Visible := True;
210   end;
211 end;
212

```