尾崎 浩司

株式会社ミガロ

システム事業部 システム2課

連携で広がるDelphi/400活用術

**開発の効率を上げるには
既存のいろいろな仕組みと「連携」することが最もよい手段だ。
ここでは、Delphi/400 と他の仕組みとを連携する手法を
具体的な手順を含め紹介する。**

略歴
1973 年 8 月 16 日 生
1996 年三重大学 工学部卒
1999 年 10 月株式会社ミガロ入社
1999 年 10 月システム事業部配属

現在の仕事内容
入社以来、主に Delphi/400 を利用した受託開発を担当している。

はじめに

Delphi/400 は、IBM i を使用するアプリケーションを開発するのに、最適な開発ツールである。と同時に、制約のない Windows アプリケーションを構築するのにも、最適な開発ツールである。

適用業務において可能性は無限に広がるが、1 から全てを構築するとなると工数がいくらあっても足りないであろう。開発の効率を上げるには、すでに用意されているいろいろな仕組みと「連携」を行うことが最適な手段だ。

本稿では、Delphi/400 と他の仕組みとを連携する手法を、具体的な手順を示しながら紹介していこう。これをきっかけに、皆様の開発のバリエーションが広がれば幸いである。

COMを使ったアプリケーション連携

「連携」というとまず思いつくのが、業務で最も使用する Excel との連携であろう。データベースから取得したデータを Excel に出力する、ソース 1 のようなアプリケーションを、私もたびたび開発している。【ソース 1】

COM

このような連携を実現するのが、「COM」と呼ばれる技術である。COM はコンポーネント・オブジェクト・モデルの略で、アプリケーションの持つ機能をオブジェクト化し、別のアプリケーションから容易に利用できるようにする技術である。

この COM という技術により、Delphi/400 からいろいろなアプリケーションの機能を使用することが可能になる。さらに、他のアプリケーションに機能を提供するプログラムを構築することも可能である。

例えば、IBM i から情報を取得するロジックを、Delphi/400 で開発し、COM とすることで、他のツールからの利用が可能になる。そうすることでエンドユーザーは、Excel や Access の VBA 等で、間接的に IBM i のデータを使用することも可能になるというわけである。

COMサーバーの開発

それでは、COM として情報を提供するサーバーはどのようにして開発するのか、見ていこう。

まず、新規プロジェクトを作成する。そして、Delphi のメニューから［ファイル］→［新規作成］→［その他］を選択し、選択カテゴリ「ActiveX」から「オートメーションオブジェクト」を選択する。【図 1a】

すると、ウィザードが始まるので、「CoClass 名」欄にクライアントがアクセスするためのクラス名を入力する。ここでは"AS400Infomation"としよう。【図 1b】

ウィザードが終了すると「タイプライブラリ」と呼ばれるインターフェースを定義する画面が表示される。【図 1c】

この画面で、外部に提供するプロパティあるいはメソッドを定義するわけだ。

"IAS400Infomation" をクリックする

図1a

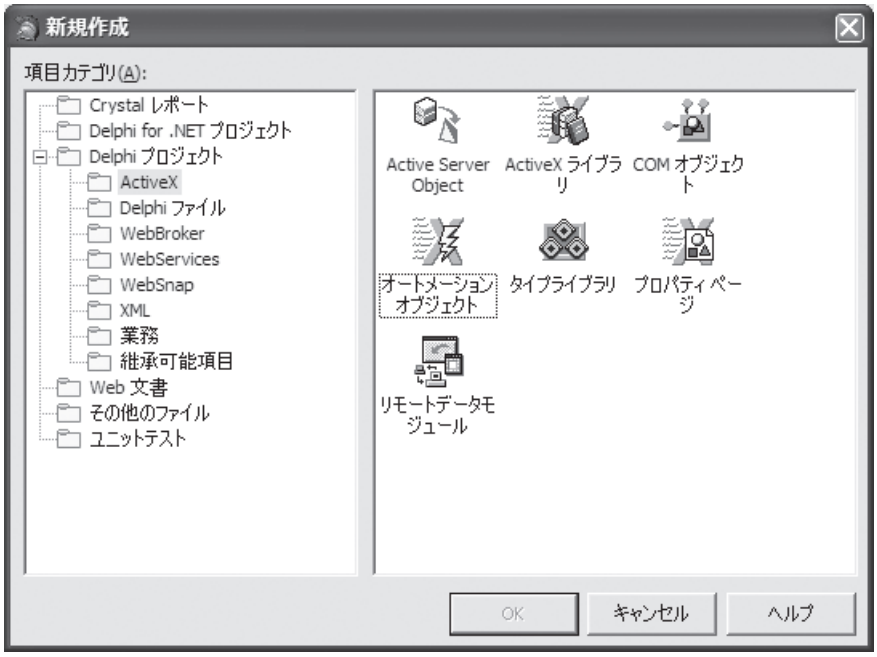

新規作成
項目カテゴリ(A):
Crystal レポート
Delphi for .NET プロジェクト
Delphi プロジェクト
ActiveX
Delphi ファイル
WebBroker
WebServices
WebSnap
XML
業務
継承可能項目
Web 文書
その他のファイル
ユニットテスト
Active Server Object
ActiveX ライブラリ
COM オブジェクト
オートメーション オブジェクト
タイプライブラリ
プロパティページ
リモートデータモジュール
OK
キャンセル
ヘルプ

図1b

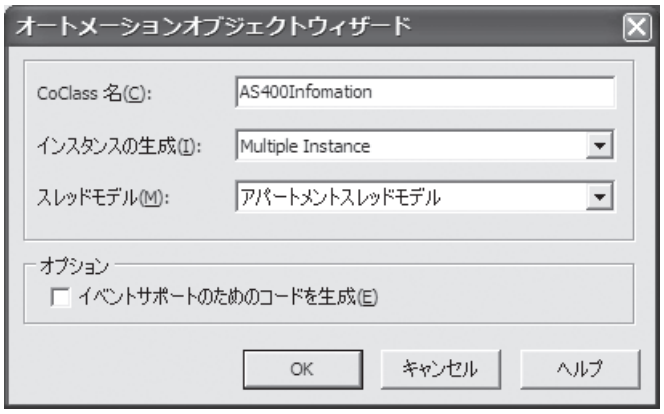
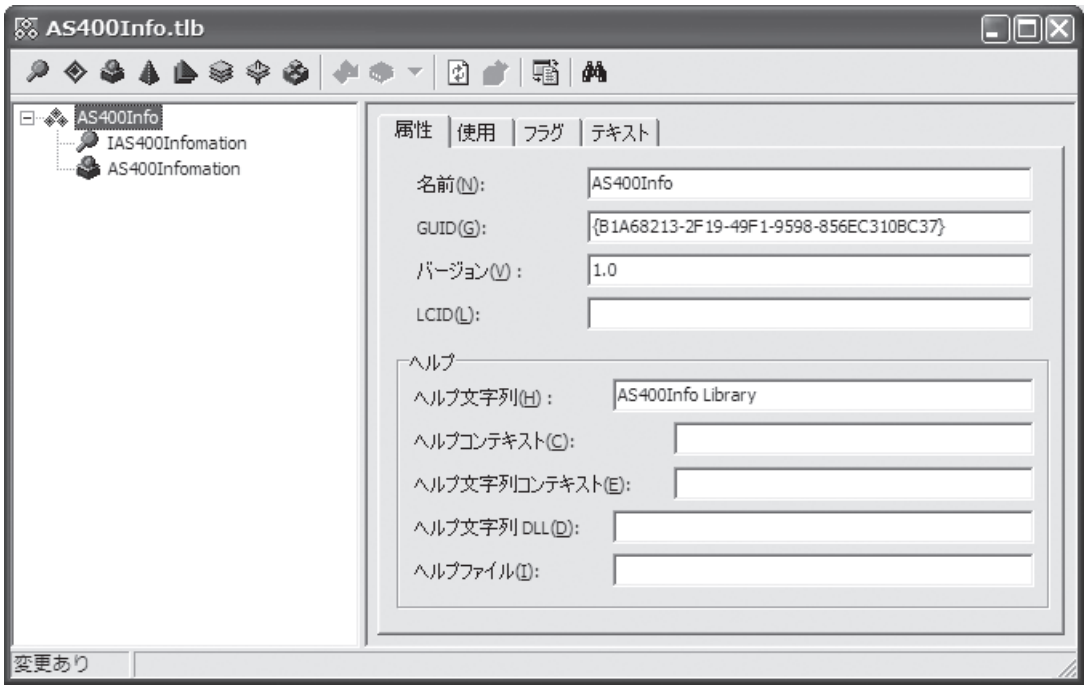

オートメーションオブジェクトウィザード
CoClass 名(C):
AS400Infomation
インスタンスの生成(I):
Multiple Instance
スレッドモデル(M):
アパートメントスレッドモデル
オプション
イベントサポートのためのコードを生成(E)
OK
キャンセル
ヘルプ

図1c


AS400Info.tlb
AS400Info
IAS400Infomation
AS400Infomation
属性
使用
フラグ
テキスト
名前(N):
AS400Info
GUID(G):
{B1A68213-2F19-49F1-9598-856EC310BC37}
バージョン(V):
1.0
LCID(L):
ヘルプ
ヘルプ文字列(H):
AS400Info Library
ヘルプコンテキスト(C):
ヘルプ文字列コンテキスト(E):
ヘルプ文字列 DLL(D):
ヘルプファイル(I):
変更あり

と、メソッドおよびプロパティの新規作成が選択可能になるので、必要に応じたインターフェース定義を行う。メソッドであれば名前を定義し、プロパティであれば名前と属性（タイプ）を定義する。

ここでは、メソッドとして"GetData"を定義し、［読み込み|書き込み］プロパティとして"CustNo"（タイプは数字属性［Long］）と［読み込み専用］プロパティとして"CustName"（タイプは文字属性［BSTR］）を定義する。【図1d】。

インターフェースの定義が終了したら、「ソースコード更新」というボタンを押してみよう。するとソース2のようなソースコードが自動作成される。あとは、そこに仕様に応じたユーザーコードを追加すればよい。【ソース2】

COMサーバーのサンプル

今回は、"CustNo"プロパティに得意先コードを指定し、GetDataメソッドを呼び出すことにより、データベースから得意先マスターを検索し、取得した得意先名を結果としてCustNameプロパティにセットするCOMサーバーを作成してみた。【ソース3】

このサンプルでは、新規プロジェクト作成時にあらかじめ用意されているフォーム（ここではfrmMainと命名）に、DataBaseコンポーネントとQueryコンポーネントを貼り付け、データベース接続定義ならびにデータ抽出用のSQLの定義を行っている。

なお、今回作成しているCOMオブジェクトは、データベースへのアクセス機能のみを持ったものとなる。そのため、画面（フォーム）は表示不要である。

このようなCOMオブジェクトを作成する場合、プロジェクトファイルのソースコードを表示し、"Application.ShowMainForm := False;"の1行を追加して完成となる。【ソース4】

COM登録

完成したプログラムを他のアプリケーションより利用可能にするためには、レジストリ登録が必要になる。

Delphi/400開発環境下でレジストリ登録を行う場合は、Delphiのメニューから［実行］→［実行時引数］を選択し、「パラメータ」欄に「/regserver」と入力し、プログラムを実行する。（反対にレジストリを解除する場合、「/unregserver」を指定して実行する）。【図2】

プログラムを実行するとどうなるか。実行後、画面は何も表示されず、そのままプログラムが終了するはずだ。ここでは、レジストリへの登録が行われるだけだからである。つまり、これでCOM登録が完了となり、他のアプリケーションから利用可能になる。

実際にこの作成したプログラムが動作するのは、COMを利用するクライアントがオブジェクトを生成したときとなる。

COMを使用するクライアント

では、COMを使用するクライアントはどうなるか。今回は、Excelから利用してみよう。

Excelを起動し、シートの中に得意先コードのセルと得意先名のセル、それから検索用のボタンを用意する。【図3】

そして、ボタンのクリックに対するイベント処理として、次のようなVBAコードを入力する。【ソース5】

完成したExcelを「マクロを有効」にして実行する。画面上で得意先コードを入力した後、検索ボタンを押すことで、データベースから検索された得意先名がセットされるだろう。【図4】

ここで、Excel上で作成したソース5のソースをよく見てみよう。言語の違いはあるが、ソース1で、Delphiから、Excelオブジェクトを生成して使用したのと似ていないか。同じように、Excelから、今回作成したCOMのオブジェクトを作成し、プロパティおよびメソッドを使用しているのがわかる。

今回は、単純なマスター検索を行う連携ではあるが、このCOMオブジェクトを使用するユーザーは、IBM iのデータ構造を意識せずにデータにアクセスできる。

つまり、IBM i上のデータをそのまま公開すると機密上よくない場合にも、このような仕組みを検討することにより、必要に応じた項目のみを公開できる。ユーザーが、IBM iのデータを、自由に安全に利用できるのではないだろうか。

それ以外にもいろいろな連携が実現できると思うので、ぜひ皆様も新たな連携にチャレンジしてほしい。

Webで提供される情報との連携

インターネットには多彩な情報が公開されている。ふだん皆様もいろいろ利用されているだろう。これらの情報とシステムとが連携すれば、便利になるのではと考えたことはないだろうか？

例えば、システムに登録されている得意先マスターの住所情報から、地図のWebサイトが表示できたり、出張精算画面で入力した駅名から路線検索ができたりするとたいへん便利と思われる。

Webサービス

世の中には一般に「Webサービス」と呼ばれる仕組みがあり、SOAPと呼ばれるプロトコルを使用すると完全な連携が可能だ。また、Delphi/400からWebサービスを使用する仕組みも用意されている。

しかし、これらを使いこなすには、SOAPやXML等の知識が必要になってくるため、少し敷居が高いのも事実である。ゆえにWebの連携は難しいと考えていないだろうか？　実は、そこまでのことをしなくても、ちょっとした連携であれば容易に実現可能である。

郵便番号検索Webアプリケーション

インターネットエクスプローラで、アドレス欄に以下のように入力してほしい。

http://search.post.japanpost.jp/cgi-zip/zipcode.php?zip=5560017

これは、日本郵便の郵便番号検索Webアプリケーションである。このURLから、郵便番号556-0017地域の所在地がわかる。【図5】

このように、一般的なWebアプリケーションは、Webブラウザから Webアプリケーションに問い合わせを行うことで動作する。

このときの問い合わせ方法には、GETメソッドとPOSTメソッドという2種類が存在するのだが、その中で

図1d

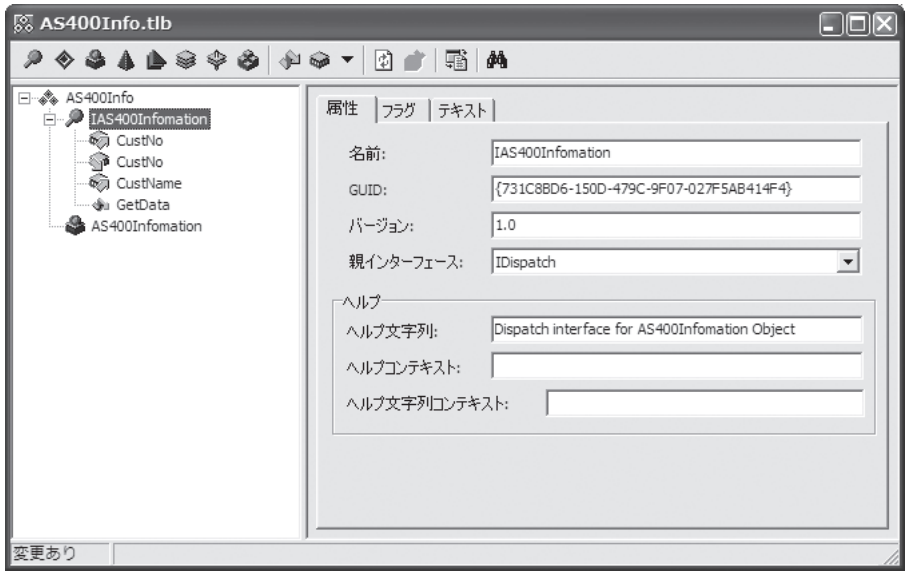

AS400Info.tlb
AS400Info
IAS400Infomation
CustNo
CustNo
CustName
GetData
AS400Infomation
属性
フラグ
テキスト
名前:
IAS400Infomation
GUID:
{731C8BD6-150D-479C-9F07-027F5AB414F4}
バージョン:
1.0
親インターフェース:
IDispatch
ヘルプ
ヘルプ文字列:
Dispatch interface for AS400Infomation Object
ヘルプコンテキスト:
ヘルプ文字列コンテキスト:
変更あり

図2

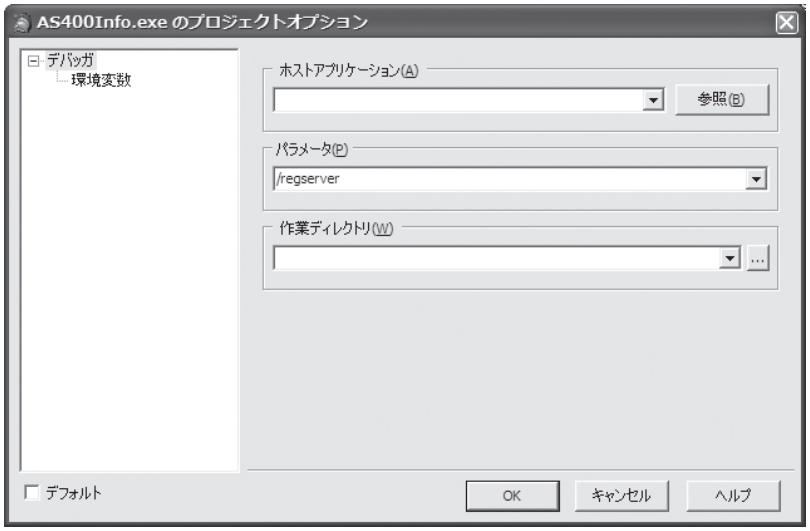
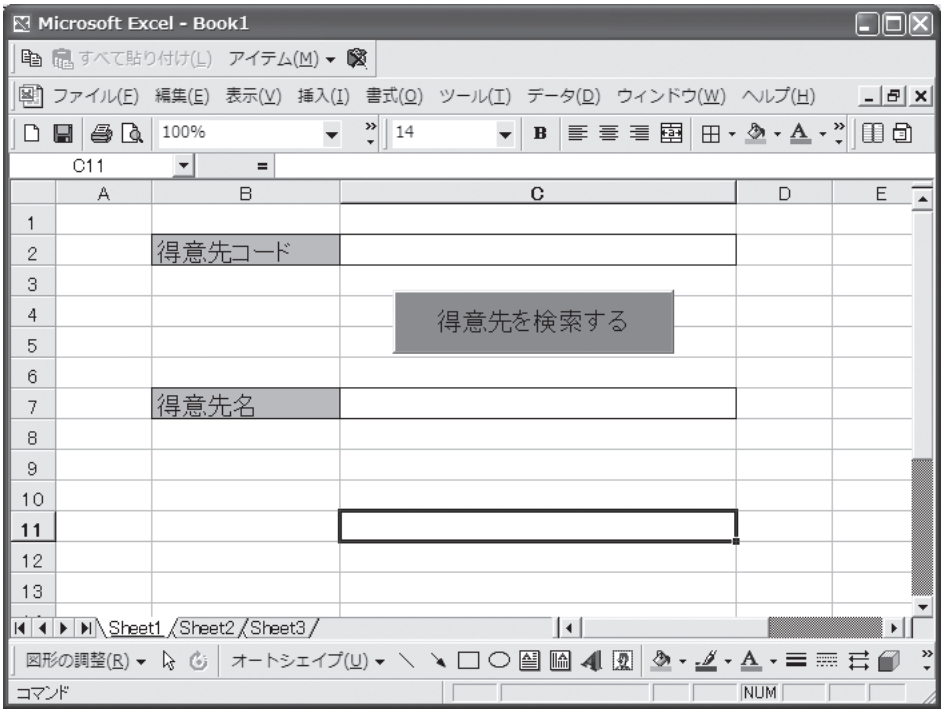

AS400Info.exe のプロジェクトオプション
デバッガ
環境変数
ホストアプリケーション(A)
参照(B)
パラメータ(P)
/regserver
作業ディレクトリ(W)
デフォルト
OK
キャンセル
ヘルプ

図3


Microsoft Excel - Book1
すべて貼り付け(L)
アイテム(M)
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)
ヘルプ(H)
100%
14
C11
得意先コード
得意先を検索する
得意先名
Sheet1
Sheet2
Sheet3
図形の調整(R)
オートシェイプ(U)
コマンド
NUM

GET メソッドは、URL の中に問い合わせを記載するのが特徴だ。先ほどの郵便番号検索の場合、"?zip=5560017" という部分が問い合わせクエリーとなる。

つまり、GET メソッドで問い合わせすることが可能な Web アプリケーションは、変数を含む URL を渡すだけで連携が可能になるのである。

連携プログラムの作成

それでは、簡単な連携プログラムを作成してみよう。

まず、新規プロジェクトを作成し、Button、Edit そして WebBrowser コンポーネントを貼り付ける。【図 6】

そして、ボタンコンポーネントの Click イベントに、処理を記述する。【ソース 6】

では、完成したプログラムを実行してみよう。先ほどのブラウザ画面からの表示結果と同じ内容が、Delphi のフォーム上に表示されることが確認できる。【図 7】

このように Web 情報への連携はとても簡単である。

Google検索

もう 1 つ見てみよう。先ほどと同じように、インターネットエクスプローラ上に下記アドレスを入力してほしい。

http://www.google.co.jp/search?num=30&q=%E3%83%9F%E3%82%AC%E3%83%AD

これは皆様おなじみの Google 検索である。検索キーワード " ミガロ " で、結果が 30 件表示されている。【図 8】

このサイトも先ほどと同じように、Get メソッドで問い合わせを行っている。"q=%E3%83…" のところを見てほしい。実は、この部分は「ミガロ」というキーワードで検索しなさいという問い合わせを表しているのだが、符号化されているのがわかるであろう。

つまり、Get メソッドで問い合わせする際には、通常 2 バイト文字等は利用できないのである。加えて、このように空白文字や特殊記号、日本語等の全角文字を符号化するルールを「URL エンコード」と呼んでいる。

では、Delphi/400 から使用する際に、URL エンコードはどうすればよいか？実は HTTPApp というユニットを uses 節に追加すると、HTTPEncode 関数が使用できるようになり、これを使うと容易に URL エンコードが可能である。

なお、URL エンコードの際には、対象の Web サイトが使用する文字コード体系によりさらに変換が必要な場合もある。先ほどの Google 検索サイトでは、UTF-8 という文字コード体系を使用しているので、このような場合、さらに AnsiToUtf8 関数を使うとよい。

Google検索を実現した Delphi/400連携プログラム

先ほど作成した日本郵便の検索プログラムを改良してみよう。ソース 7 は、Google 検索を実現した Delphi/400 連携プログラムとなる。【ソース 7】

完成したプログラムを実行すると、Delphi の画面で指定したキーワードをもとに Google 検索を行い、結果が画面に表示されていることがわかる。【図 9】

さいごに

このように、単純に Web サイトに対して問い合わせし、結果を画面に表示するだけの「連携」であった。とはいえ、皆様が開発するアプリケーションにおいて、入力した値がそのままパラメータとして利用できるようになれば、いろいろな呼び出しが可能になるだろう。

今回紹介した以外にも、Get メソッドを使用した検索可能なサイトが多数存在する。

例えば、http://ready.to/search/list/ というサイトでは、ブラウザから直接呼び出せるサイトが紹介されている。参考にしてはいかがだろうか。

アイデアしだいでは便利な連携が可能と思われるので、ぜひともいろいろチャレンジしていただきたい。

M

図4

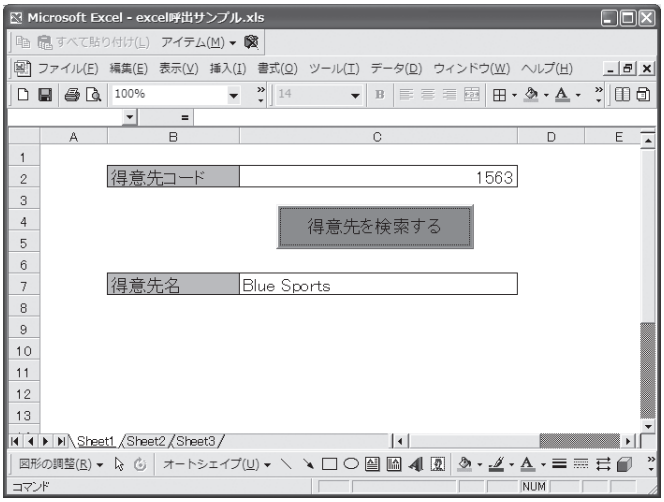

Microsoft Excel - excel呼出サンプル.xls
得意先コード
1563
得意先を検索する
得意先名
Blue Sports
Sheet1
Sheet2
Sheet3

図5


郵便番号 5560017 の検索結果 - 日本郵便 - Windows Internet Explorer
日本郵便
あたらしい ふつうをつくる。
郵便番号から検索
郵便番号 5560017 の検索結果
大阪府
マ行
556-0017
大阪府
大阪市浪速区
湊町
ミナトマチ

図6

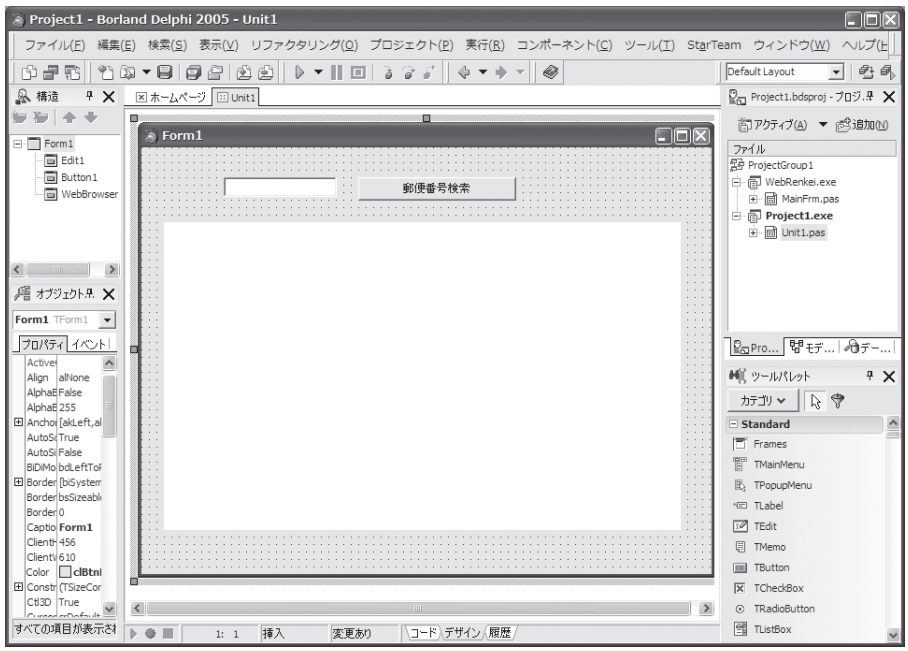

Project1 - Borland Delphi 2005 - Unit1
Form1
郵便番号検索

図7

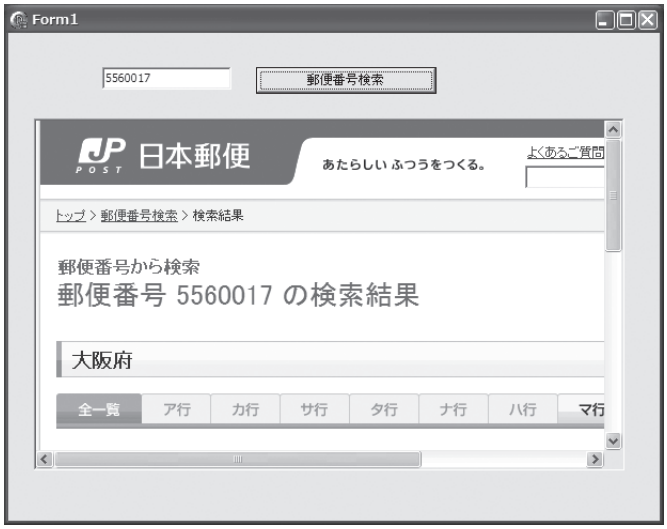
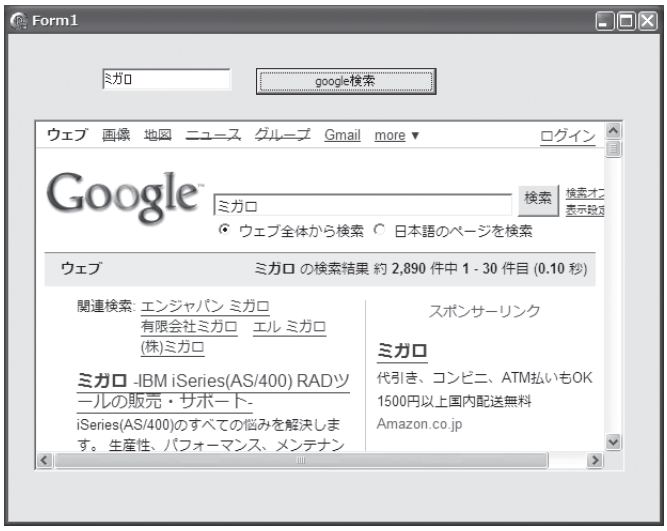

Form1
5560017
郵便番号検索
日本郵便
あたらしい ふつうをつくる。
よくあるご質問
トップ > 郵便番号検索 > 検索結果
郵便番号から検索
郵便番号 5560017 の検索結果
大阪府
全一覧
ア行
カ行
サ行
タ行
ナ行
ハ行
マ行

図8


ミガロ - Google 検索 - Windows Internet Explorer
http://www.google.co.jp/search?num=30&q=%E3%83%9F%E3%82%AC%E3%83%AD
Live Search
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)
ミガロ - Google 検索
ページ(P) ツール(O)
ウェブ 画像 地図 ニュース グループ Gmail more ▼
ログイン
Google
ミガロ
検索
検索オプション
表示設定
ウェブ全体から検索 日本語のページを検索
ウェブ
ミガロ の検索結果 約 2,890 件中 1 - 30 件目 (0.20 秒)
関連検索: エンジャパン ミガロ 有限会社ミガロ エル ミガロ (株)ミガロ
ミガロ -IBM iSeries(AS/400) RADツールの販売・サポート-
iSeries(AS/400)のすべての悩みを解決します。 生産性、パフォーマンス、メンテナンス性に優れたDelphi/400、webアプリケーション開発の決定版JACi400でiSeriesの性能が120%up！
www.migaro.co.jp/ - 24k - キャッシュ - 関連ページ
Delphi/400 Version2007 発売開始
MIGARO＞導入事例＞事例一覧
noMAX特長
MKS特長
Tips
TecAward
migaro.co.jp からの検索結果 »
[PDF] 株式会社ミガロ 「第18期を迎えて」 社長 上甲 將隆
ファイルタイプ: PDF/Adobe Acrobat - HTMLバージョン
ミガロには「クレド」がある。会社の理念というよりは、皆で大切にすること、皆で伸ばしていきたい資質が書いてある ... さらに近年では、お客様の要望や問題が多様化しミガロ 単独ではなくパートナーやベンダーの方々とタッグを組んで提案する機会も増え ...
www.migaro.co.jp/contents/company/newspaper/images/MIGZINEVol6.pdf - 関連ページ
スポンサーリンク
ミガロ
代引き、コンビニ、ATM払いもOK
1500円以上国内配送無料
Amazon.co.jp
ページが表示されました
インターネット
100%

図9


Form1
ミガロ
google検索
ウェブ 画像 地図 ニュース グループ Gmail more ▼
ログイン
Google
ミガロ
検索
ウェブ全体から検索 日本語のページを検索
ウェブ
ミガロ の検索結果 約 2,890 件中 1 - 30 件目 (0.10 秒)
関連検索: エンジャパン ミガロ
有限会社ミガロ エル ミガロ
(株)ミガロ
ミガロ -IBM iSeries(AS/400) RADツールの販売・サポート-
iSeries(AS/400)のすべての悩みを解決します。 生産性、パフォーマンス、メンテナン
スポンサーリンク
ミガロ
代引き、コンビニ、ATM払いもOK
1500円以上国内配送無料
Amazon.co.jp

ソース1

```
uses ComObj;

procedure TForm1.Button1Click(Sender: TObject);
var
  MsExcel, MsApplication, WBook, WSheet: OleVariant;
  i: integer;
begin
  //Excel 起動
  MsExcel := CreateOleObject('Excel.Application');
  MsApplication := MsExcel.Application;
  MsApplication.Visible := True;
  WBook  := MsApplication.WorkBooks.Add;
  WSheet :=WBook.ActiveSheet;
  //Excel にタイトル出力
  WSheet.Cells[1,3].Value := ' 得意先マスター一覧表 ';
  WSheet.Cells[1,3].Font.Size := 15;
  WSheet.Cells[2,1].Value := ' 得意先コード ';
  WSheet.Cells[2,2].Value := ' 得意先名 ';
  WSheet.Cells[2,3].Value := ' 住所 ';
  WSheet.Cells[2,1].Font.Bold := 'True';
  WSheet.Cells[2,2].Font.Bold := 'True';
  WSheet.Cells[2,3].Font.Bold := 'True';
  //Excel にデータ出力
  with Table1 do
  begin
    Active := True;
    try
    i := 0;
    First;
    while not eof do
    begin
      WSheet.Cells[i+3,1].Value := FieldByName('CUSTNO').
      AsInteger;
      WSheet.Cells[i+3,2].Value :=
      FieldByName('COMPANY').AsString;
      WSheet.Cells[i+3,3].Value := FieldByName('ADDR1').
      AsString;
      Next;
      i := i + 1;
    end;
    finally
      Active := False;
    end;
  end;
end;
```

ソース2

```
unit Unit1;

{$WARN SYMBOL_PLATFORM OFF}

interface

uses
  ComObj, ActiveX, AS400Info_TLB, StdVcl;

type
  TAS400Infomation = class(TAutoObject,
  IAS400Infomation)
  protected
    function Get_CustName: WideString; safecall;
    function Get_CustNo: Integer; safecall;
    procedure GetData; safecall;
    procedure Set_CustNo(Value: Integer); safecall;

  end;

implementation

uses ComServ;

function TAS400Infomation.Get_CustName: WideString;
begin

end;

function TAS400Infomation.Get_CustNo: Integer;
begin

end;

procedure TAS400Infomation.GetData;
begin

end;

procedure TAS400Infomation.Set_CustNo(Value: Integer);
begin

end;

initialization
  TAutoObjectFactory.Create(ComServer,
  TAS400Infomation, Class_AS400Infomation,
    ciMultiInstance, tmApartment);
end.
```

ソース3

```
----- << ここまで省略 >> ------

type
 TAS400Infomation = class(TAutoObject,
 IAS400Infomation)
 private
  // 内部変数
  FCustNo: Integer;          // 得意先コード
  FCustName: String;          // 得意先名
 protected
  function Get_CustName: WideString; safecall;
  function Get_CustNo: Integer; safecall;
  procedure GetData; safecall;
  procedure Set_CustNo(Value: Integer); safecall;
 end;

implementation

uses ComServ, MainFrm, SysUtils, Dialogs, DBTables;

function TAS400Infomation.Get_CustName: WideString;
begin
 // 得意先名の内部値を渡す
 Result := FCustName;
end;

function TAS400Infomation.Get_CustNo: Integer;
begin
 // 得意先コードの内部値を渡す
 Result := FCustNo;
end;

procedure TAS400Infomation.GetData;
begin
 try
  // 得意先コードが指定されない場合、エラーとする
  if FCustNo = 0 then
   raise Exception.Create(' 得意先コードが指定されていま
   せん。');

  // クエリーを使用し、得意先コードをキーに得意先マス
  ターを検索し、
  // 得意先名を取得する
  with frmMain.Query1 do
  begin
   // クエリーを閉じる
   Active := False;

   // 得意先コードパラメータに得意先コード内部値を代入
   する
   ParamByName('CUSTNO').AsInteger := FCustNo;

   // クエリーを実行する
   Active := True;
   try
    First;

    // データが存在しない場合、エラーとする
    if Eof and Bof then
     raise Exception.Create(' 指定された得意先コードは
     存在しません。');

    // 取得結果を得意先名内部値に代入する
    FCustName := FieldByName('CUSTNM').AsString;
   finally
    // クエリーを閉じる
    Active := False;
   end;
  end;
 except
  // エラーメッセージの表示
  on E: Exception do
  begin
   MessageDlg(E.Message, mtError, [mbOK], 0);
  end;
 end;
end;

procedure TAS400Infomation.Set_CustNo(Value: Integer);
begin
 // パラメータを内部値にセットし、得意先名を初期化する
 FCustNo := Value;
 FCustName := '';
end;

----- << 以下省略 >> -----
```

ソース4

```
Application.Initialize;
Application.CreateForm(TfrmMain, frmMain);
Application.ShowMainForm := False;  // ( 追加 ) メイン
フォームを表示しない
Application.Run;
```

ソース5

```
Option Explicit


Private Sub CommandButton1_Click()

  Dim objAS400Info As Object   '---- 得意先情報取得オブ
  ジェクト
  Dim lngCustNo As Long        '---- 得意先コード

  ' 得意先情報取得オブジェクトの生成
  Set objAS400Info = CreateObject("AS400Info.
  AS400Infomation")

  ' 取引先コードのセット
  lngCustNo = ActiveSheet.Range("C2").Value
  objAS400Info.CustNo = lngCustNo

  ' データの取得 ( メソッドの実行）
  objAS400Info.GetData

  ' 取得した得意先名のセット
  ActiveSheet.Range("C7").Value = objAS400Info.
  CustName

  ' オブジェクトの解放
  Set objAS400Info = Nothing

End Sub
```

ソース6

```
procedure TForm1.Button1Click(Sender: TObject);
var
  sURLText: String;
begin
  // URL 文の作成
  sURLText := 'http://search.post.japanpost.jp/cgi-zip/
  zipcode.php?zip=';
  sURLText := sURLText + Edit1.Text;

  // 問い合わせを実行し、結果を WebBrowser コンポーネン
  トに表示
  WebBrowser1.Navigate(sURLText);
end;
```

ソース7

```
uses HTTPApp;  // HTTPApp ユニットを追加する

procedure TForm1.Button1Click(Sender: TObject);
var
  sURLText: String;
begin
  // URL 文の作成
  sURLText := 'http://www.google.co.jp/
search?num=30&q=';
  sURLText := sURLText + HTTPEncode(AnsiToUtf8(Edit1.
  Text));

  // 問い合わせを実行し、結果を WebBrowser コンポーネン
  トに表示
  WebBrowser1.Navigate(sURLText);
end;
```