

【セッションNo. 4】

Delphi/400開発ノウハウお教えします 「Googleマップ連携によるリッチなGUIアプリ開発」

株式会社ミガロ.
システム事業部 プロジェクト推進室
小杉 智昭

【アジェンダ】

Webサービス概要

Webサービスを利用するには

Google マップを使ったアプリケーション例

■ はじめに

2000年代初めごろに登場した「Webサービス」は着々と利用が広がり、**さまざまなサービスが提供**されるようになりました。この「Webサービス」は、Web関連のプログラムだけでなく、クライアントPC上で稼働する**VCLフォームアプリケーション**（以下、**クライアントアプリ**と呼称します）**でも利用**できます。

今回はWeb上で広く利用されている「Googleマップ」を中心に幾つかの「Webサービス」を**クライアントアプリ上で連携**させ、**一歩進んだアプリケーション**を作成してみましよう。

■ Webサービス概要

• APIが公開されているWebサービス

WAFL(Web API Fan's Library)というサイトによるとAPI提供団体は100を超え、登録API数は205件にもものぼります。提供の形態(有料・無料、商用利用の可・不可)はさまざまですので、**各サービスの利用規約を必ずご確認ください。**

ワッフル/WAFL (URL: <http://wafl.net/>)

WAFLに登録されているAPIをいくつかピックアップしますと...

- Google Maps(グーグル株式会社)
- Amazon AWS(Amazon.com, Inc.)
- Weather Hacks(株式会社サイブリア)
- Bing API(マイクロソフト株式会社)
- 国立国会図書館サーチAPI(国立国会図書館)



MTS2011 Sample Form
[Close] [Maximize] [Minimize]

取引先CD
事業所CD
取引先名
事業所名
郵便番号
住所1
住

| | | | | | | |
|---------|---|----------|-------|---------|-------------------|---|
| 10000 | 1 | 株式会社ミガロ. | 東京事業所 | 1060041 | 東京都港区麻布台1-4-3 | 工 |
| ▶ 10000 | 2 | 株式会社ミガロ. | 大阪本社 | 5560017 | 大阪府大阪市浪速区湊町2-1-57 | 業 |

緯度 経度 位置保存

最寄駅検索



JR難波(143m)【JR関西本線(大和路)】

<最寄駅未選択>

JR難波(143m)【JR関西本線(大和路線)(加茂~JR難波)】

なんば(239m)【大阪市営地下鉄四つ橋線】

大阪難波(405m)【阪神なんば線】

大阪難波(405m)【近鉄難波線】

なんば(446m)【大阪市営地下鉄千日前線】

なんば(463m)【大阪市営地下鉄御堂筋線】

■ Webサービス概要

• Webサービスについての注意

通常、Webサービスは蓄積したデータを加工して返すだけのシンプルなサービスになります。

最寄駅検索の場合...

Q) 緯度・経度をパラメータとして渡す

A) 一定範囲内に存在する駅情報を近いものから順にXMLやJSONといったテキストデータにして返す

→ Googleやマイクロソフト等が実施している地図そのものを
インターフェースごと提供するサービスは稀

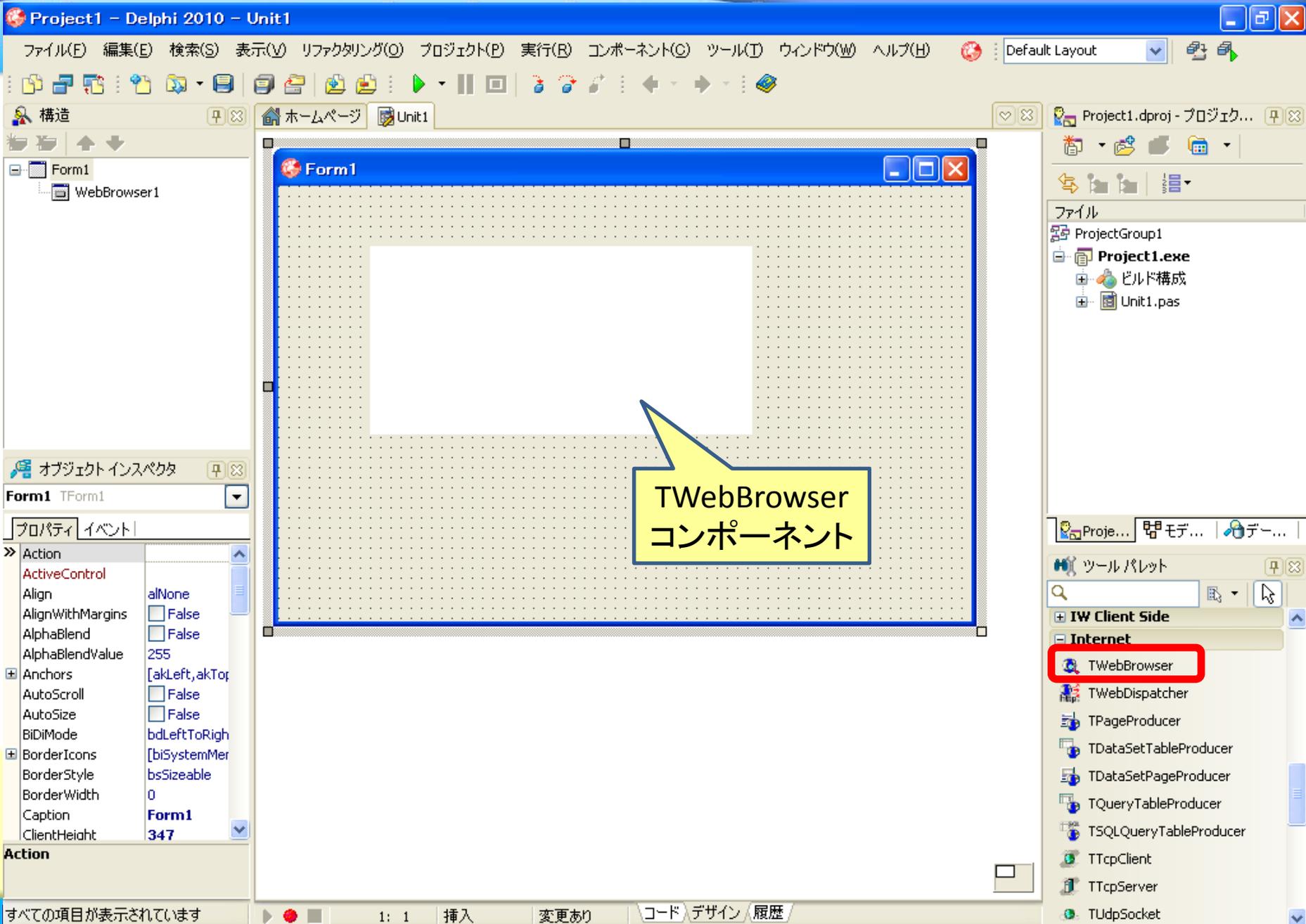
■ Webサービス概要

• 以前に紹介したものとの違い (Google Maps)

Googleの地図表示については、第2回テクニカルセミナーでご紹介済み！？

(以前) Googleの地図ページに検索ワードを与えて表示
(今回) APIを使って、地図そのものを直接制御





TWebBrowser
コンポーネント

TWebBrowser

■ Webサービスを利用するには

• Google Mapsを利用するには

TWebBrowserで任意のページを開く方法

① Webページを開く

例: `WebBrowser1.Navigate('http://www.migaro.co.jp');`

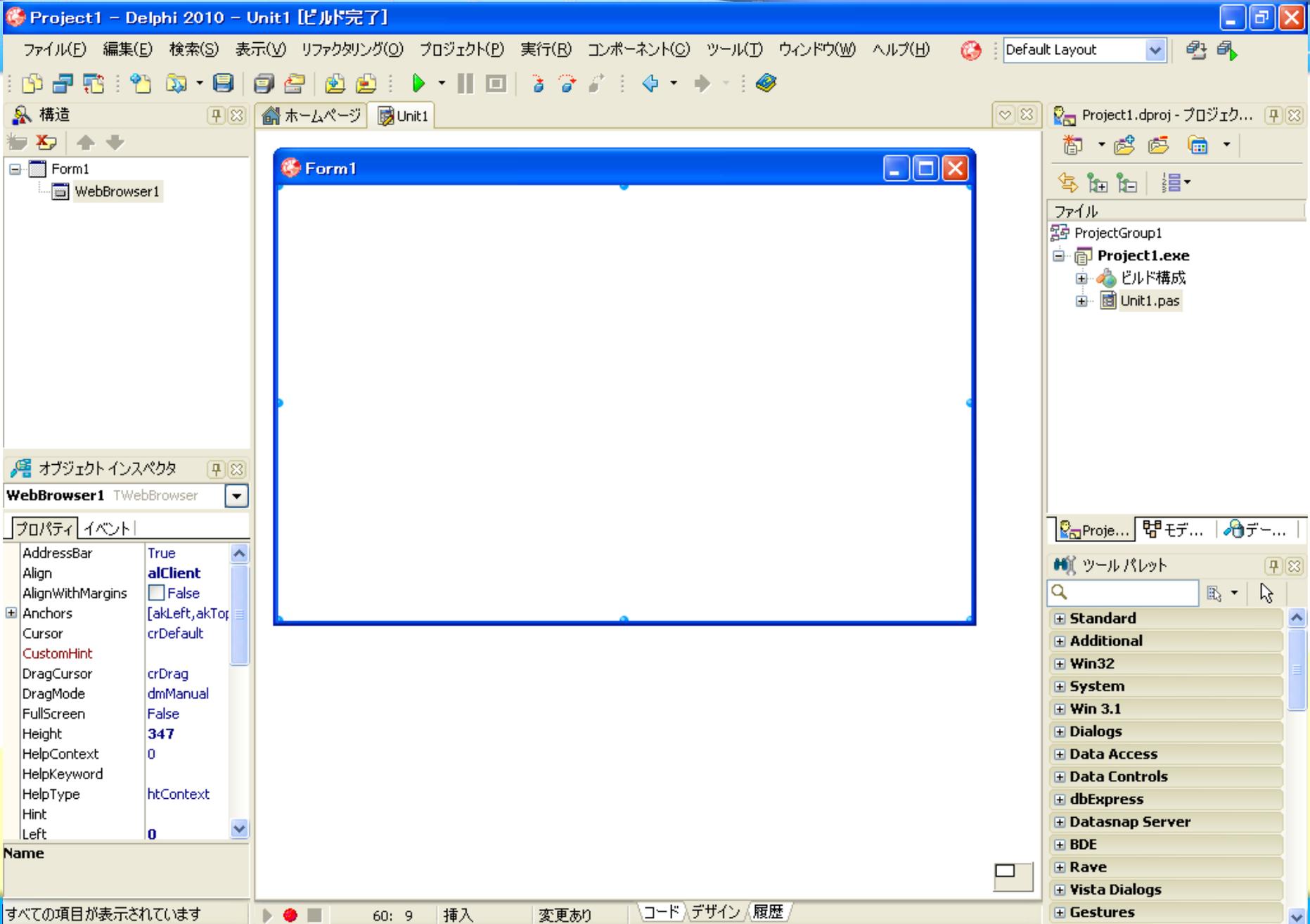
② ローカルのファイルを開く

例: `WebBrowser1.Navigate('file:///c:¥temp¥test.html');`

③ プログラム内で指定したHTMLデータを開く

参考: How to load HTML directly to a WebBrowser

http://delphi.about.com/cs/adptips2004/a/bltip0104_4.htm



■ Webサービスを利用するには

- Google Mapsを利用するには
TWebBrowserでGoogleマップを表示し

センサーは
使用しない

```
<html>
<head>
<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
<script type="text/javascript" src="http://maps.google.com/maps/api/js?sensor=false"></script>
<script type="text/javascript">
  function initialize() {
    var latlng = new google.maps.LatLng(35.66271265557856, 139.73858686381533)
    var myOptions = {
      zoom: 15,
      center: latlng,
      mapTypeId: google.maps.MapTypeId.ROADMAP
    };
    var map = new google.maps.Map(document.getElementById("map_c
  }
</script>
</head>
<body onload="initialize()">
  <div id="map_canvas" style="width:100%; height:100%">
</div>
</body>
</html>
```

弊社、東京事業所の
緯度、経度

■ Webサービスを利用するには

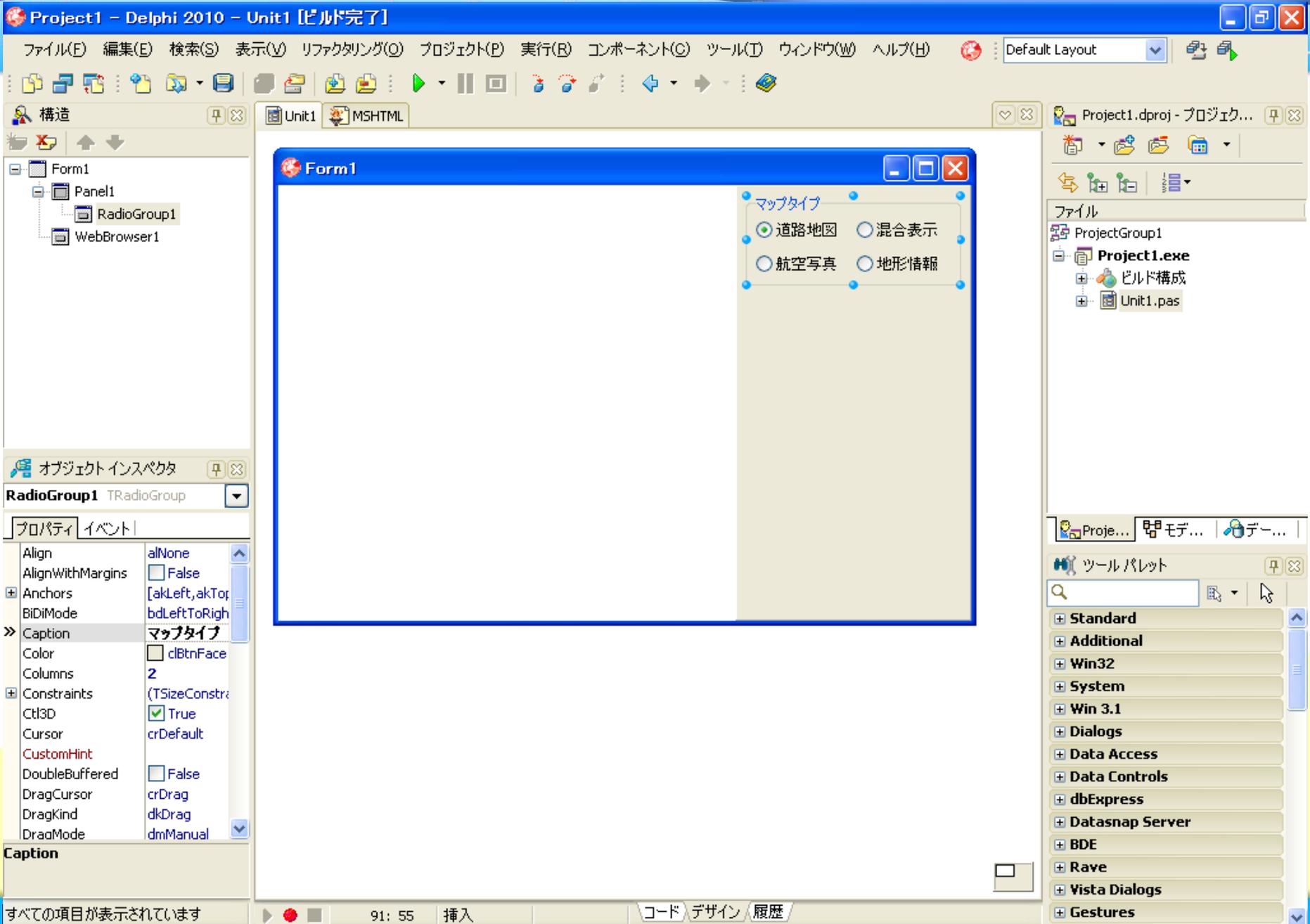
• Google Mapsを利用するには

TWebBrowserでGoogleマップを表示してみましょう。

```
procedure TForm1.Button1Click(Sender: TObject);
var
  sl: TStr
  ms: TMem
begin
  WebBrows
  while We
  Applic

  if Assig
  begin
    sl :=
    try
      ms :
      try
        sl
        sl
        ms
        (W
      fina
      ms
    end;
  finally
    sl.F
  end;
end;
end;
```

Create(ms));

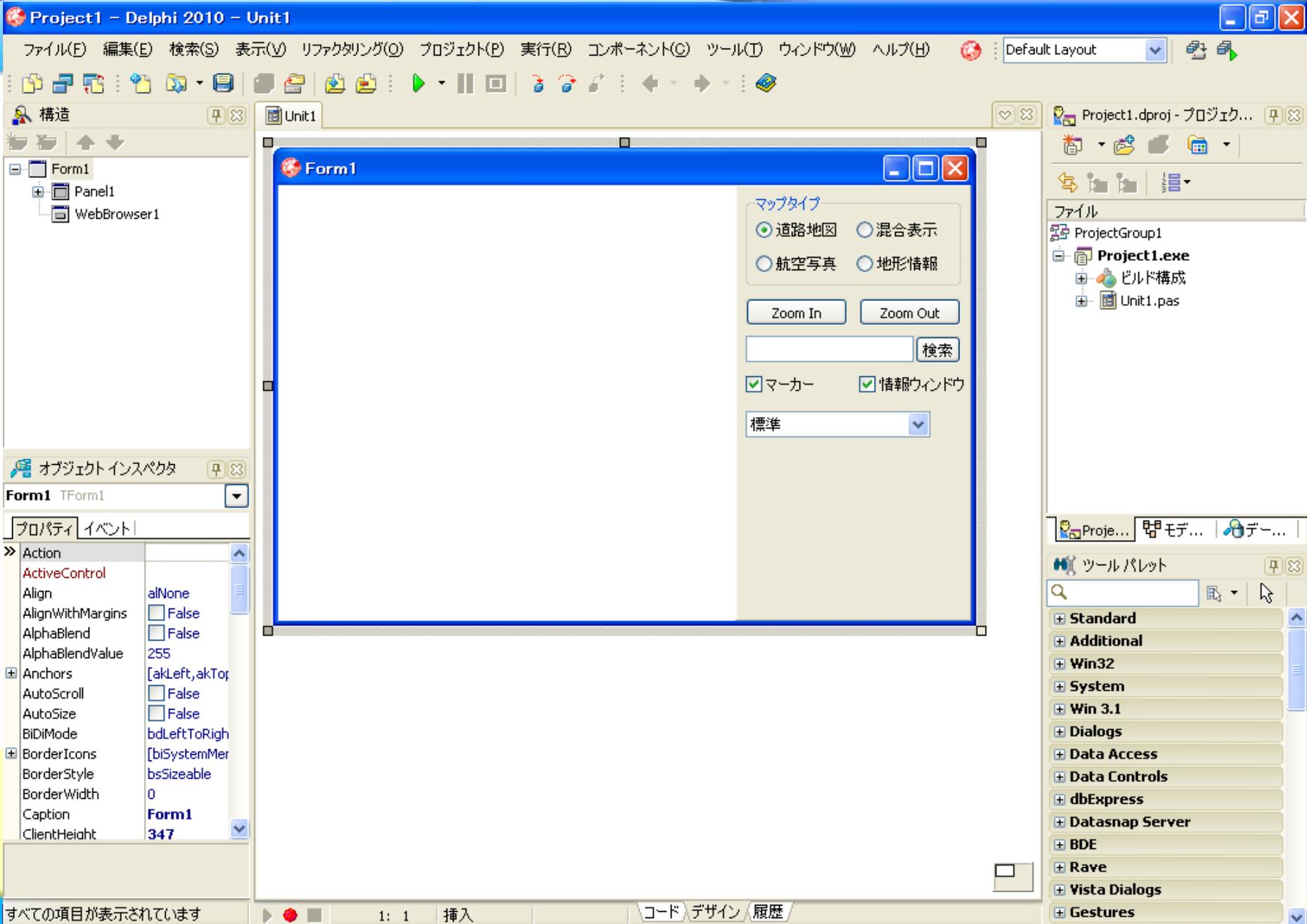


■ Webサービスを利用するには

●

The image shows three overlapping browser windows, each displaying a Google Maps interface. The windows are titled 'Form1'. The top-left window shows the '道路地図' (Road Map) style selected. The middle window shows the '混合表示' (Hybrid) style selected. The bottom-right window shows the '地形情報' (Terrain) style selected. The 'マップタイプ' (Map Type) menu is visible in each window, with the selected option highlighted. The map content shows a street view of a city area with various landmarks and buildings.

戻しましょう。



■ Webサービスを利用するには

• Google Mapsを利用するには

Google Maps APIが提供する機能を幾つか確認してみましょう。

①地図を表示する倍率を変更する (Mapクラス)

```
procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
    // Zoom In
    ExecJS('map.setZoom(map.getZoom() + 1)');
end;

procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
    // Zoom Out
    ExecJS('map.setZoom(map.getZoom() - 1)');
end;
```

■ Webサービスを利用するには

• Google Mapsを利用するには

Google Maps APIが提供する機能をいくつか確認してみましょう。

②指定した住所の地図を表示する(Geocoderクラス)

```
function gotoAddress(address) {
    geocoder.geocode({"address": address}, function(results, status) {
        if (status == google.maps.GeocoderStatus.OK) {
            map.setCenter(results[0].geometry.location);
        } else {
            alert("住所検索が失敗しました。 \nステータス: " + status);
        }
    });
}
```

```
procedure TForm1.SpeedButton3Click(Sender: TObject);
begin
    // 住所検索
    ExecJS('gotoAddress("' + Edit1.Text + '")');
end;
```

■ Webサービスを利用するには

• Google Mapsを利用するには

Google Maps APIが提供する機能を幾つか確認してみましょう。

③各種制御ウィンドウを制御する

(Markerクラス / InfoWindowクラス)

```
procedure TForm1.CheckBox1Click(Sender: TObject);
begin
    // マーカー制御
    if CheckBox1.Checked then
        ExecJS('marker.setVisible(true)')
    else
        ExecJS('marker.setVisible(false)');
end;

procedure TForm1.CheckBox2Click(Sender: TObject);
begin
    // 情報ウィンドウ制御
    if CheckBox2.Checked then
        ExecJS('infowindow.open(map, marker)')
    else
        ExecJS('infowindow.close()');
end;
```

■ Webサービスを利用するには

• Google Mapsを利用するには

Google Maps APIが提供する機能をいくつか確認してみましょう。





構造

- Panel1
 - CheckBox1
 - CheckBox2
 - ComboBox1
 - Edit1
 - Edit2
 - Edit3
 - Label1
 - Label2
 - RadioGroup1
 - SpeedButton1
 - SpeedButton2
 - SpeedButton3

オブジェクトインスペクタ

Edit2 TEdit

プロパティ イベント

| | |
|--------------------|--|
| ParentColor | <input type="checkbox"/> False |
| ParentCtl3D | <input checked="" type="checkbox"/> True |
| ParentCustomHint | <input checked="" type="checkbox"/> True |
| ParentDoubleBuffer | <input checked="" type="checkbox"/> True |
| ParentFont | <input checked="" type="checkbox"/> True |
| ParentShowHint | <input checked="" type="checkbox"/> True |
| PasswordChar | #0 |
| PopupMenu | |
| ReadOnly | <input type="checkbox"/> False |
| ShowHint | <input type="checkbox"/> False |
| TabOrder | 5 |
| TabStop | <input checked="" type="checkbox"/> True |
| Tag | 0 |
| Text | |
| TextHint | |

Text

ホームページ Unit1 GoogleMap

Form1

マップタイプ

道路地図 混合表示

航空写真 地形情報

Zoom In Zoom Out

検索

マーカー 情報ウィンドウ

標準

緯度

経度

Project1.dproj - プロジェク...



ファイル

- ProjectGroup1
 - Project1.exe
 - ビルド構成
 - GoogleMap.pas
 - Unit1.pas

Proje... モデ... デー...

ツールパレット

- Standard
- Additional
- Win32
- System
- Win 3.1
- Dialogs
- Data Access
- Data Controls
- dbExpress
- Datasnap Server
- BDE
- Rave
- Vista Dialogs
- Gestures

■ Webサービスを利用するには

• Google Mapsを利用するには

Google マップ上で行われた操作をクライアントアプリ側に通知してみましょう。

```
google.maps.event.addListener(map, "click",  
    function(event) {  
        if (event.latLng) {  
            external.JSValue = event.latLng;  
        }  
    }  
);
```

■ Webサービスを利用するには

• Google Mapsを利用するには

Googleマップ上で行われた操作をクライアントアプリ側に

Unit1; GoogleMap

```
unit Unit1;  
interface  
uses  
  Windows, Messages,  
  Dialogs, OleCtrls,  
  
private  
  { Privat  
  procedur  
public  
  { Public
```

Form1

マップタイプ
 道路地図 混合表示
 航空写真 地形情報

Zoom In Zoom Out

検索

マーカー 情報ウィンドウ

標準

緯度 35.66548454463897
経度 139.73920913630673

```
WebBrowser1.UnSetJSValue := SetJSValue;  
end;
```

Project1 - Delphi 2010 - Unit1

ファイル(F) 編集(E) 検索(S) 表示(V) リファクタリング(O) プロジェクト(P) 実行(R) コンポーネント(C) ツール(T) ウィンドウ(W) ヘルプ(H)

Default Layout

構造

ホームページ Unit1

Form1

- IdHTTP1
- XMLDocument1

Form1

IdHTTP1 XMLDocument1

[Indy Clients]カテゴリ
TIdHTTP
コンポーネント

[Internet]カテゴリ
TXMLDocument
コンポーネント

XMLDocument1 TXMLDOCUMENT

プロパティ イベント

| | |
|---------------|--------------------------------|
| Active | <input type="checkbox"/> False |
| DOMVendor | MSXML |
| FileName | |
| >> Name | XMLDocument1 |
| NodeIndentStr | <空白 2個> |
| Options | [doNodeAutoCre |
| ParseOptions | [] |
| Tag | 0 |
| XML | (xml) |

Name

すべての項目が表示されています

1: 1 挿入 変更あり コード デザイン 履歴

Project1.dproj - プロジェク...

ファイル

- ProjectGroup1
- Project1.exe
- ビルド構成
- Unit1.pas

Proje... モデ... デー...

ツールパレット

Internet

- TWebBrowser
- TWebDispatcher
- TPageProducer
- TDataSetTableProducer
- TDataSetPageProducer
- TQueryTableProducer
- TSQLQueryTableProducer
- TTcpClient
- TTcpServer
- TUdpSocket
- TXMLDocument

InternetExpress

WebSnap

WebServices

■ Webサービスを利用するには

• SimpleAPI最寄駅APIを利用するには

SimpleAPI vol.2 - 最寄り駅Webサービスについて

URL: <http://map.simpleapi.net/>

SimpleAPI vol.2 最寄り駅Webサービス

SimpleAPI BETA

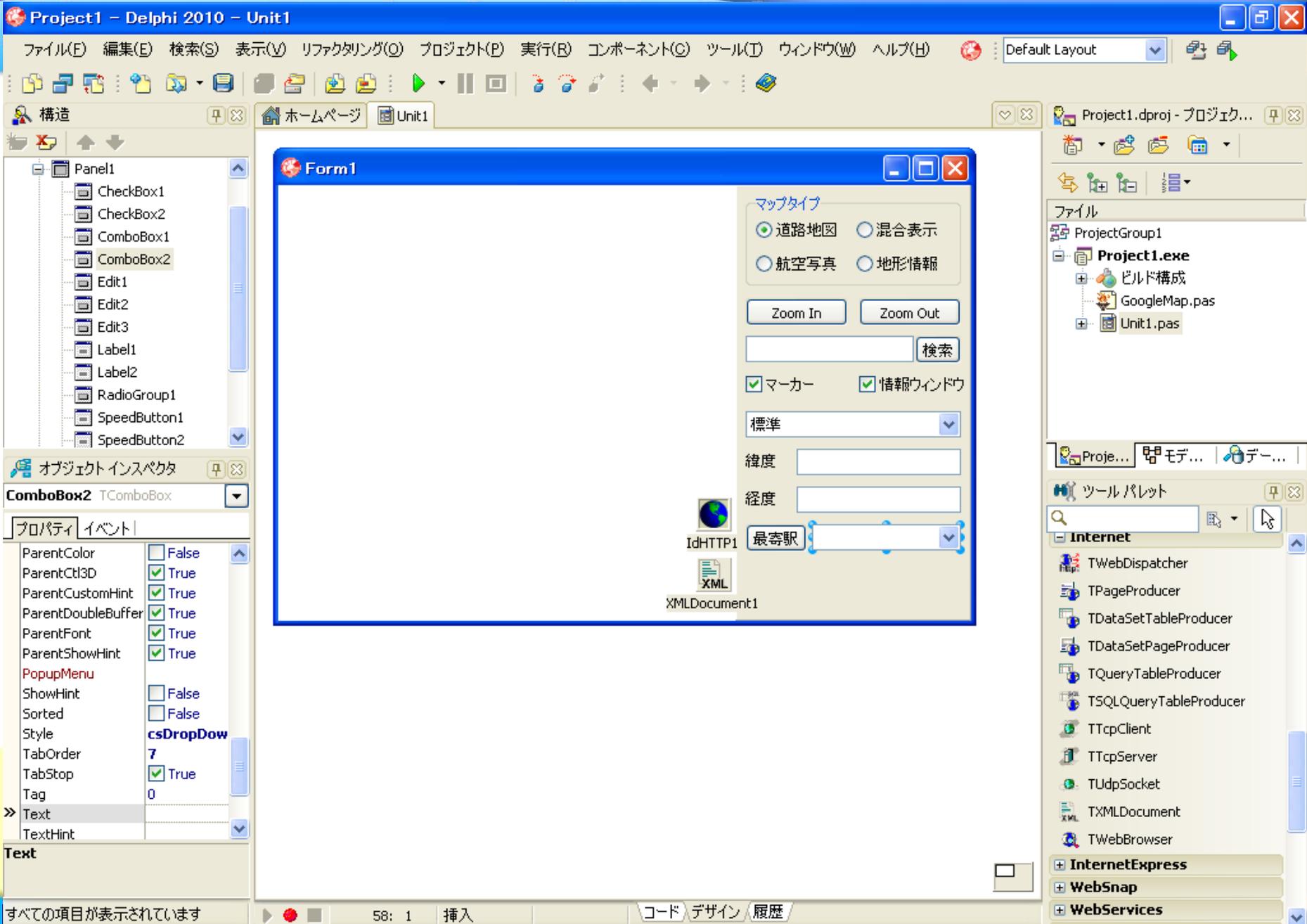
SimpleAPI「最寄り駅Webサービス」とは？

経緯度データを指定すると、その付近にある駅(もより駅)の駅名や方角などを表示するウェブAPI(Web Service)です。たとえば、東京ドーム(東京都文京区後楽1-3)の経緯度データ(x, yで指定)を渡すと、後楽園駅や水道橋駅といった**最寄り駅のデータ**や**最寄り駅地図**を返します。

「この場所って駅からどれくらい？どっち？」というのを調べてくれるAPIです。最近ではGoogleMapsなど地図からみのAPIが流行っているのでそれと合わせて利用すると、いっそう便利になります。

SimpleAPIシリーズの特徴でもある**画像方式のAPI**やJavascriptやHTML出力も用意しており、プログラム無しで最寄り駅地図画像を表示することができます(※1ページあたり最大10枚まで)。





■ Webサービスを利用するには

• SimpleAPI最寄駅APIを利用するには

Googleマップから取得した緯度・経度を使って、最寄駅を検索してみましょう。

```
procedure TForm1.SpeedButton4Click(Sender: TObject);
const
  cRequestURL = 'http://www.ekidata.jp/api/s.php?lat=%s&lon=%s';
  cFmtStation = '%s(%sm), [%s]';
  cFmtLocation = '%s, %s';
var
  i: Integer;
begin
  // 最寄駅抽出
  XMLDocument1.XML.Text := IdHTTP1.Get(Format(cRequestURL, [Edit2.Text, Edit3.Text]));
  XMLDocument1.Active := True;
  try
    ComboBox2.Clear;
    FLocation.Clear;
    if XMLDocument1.DocumentElement.ChildNodes.FindNode('coord').ChildNodes.FindNode('error').NodeValue = '0' then
    begin
      ComboBox2.Items.Add('<最寄駅未選択>');
      FLocation.Add('0, 0');
      for i := 0 to XMLDocument1.DocumentElement.ChildNodes.Count - 1 do
      begin
        if XMLDocument1.DocumentElement.ChildNodes.Nodes[i].NodeName = 'station' then
        begin
          ComboBox2.Items.Add(Format(cFmtStation, [
            XMLDocument1.DocumentElement.ChildNodes.Nodes[i].ChildNodes.FindNode('station_name').NodeValue,
            XMLDocument1.DocumentElement.ChildNodes.Nodes[i].ChildNodes.FindNode('meter').NodeValue,
            XMLDocument1.DocumentElement.ChildNodes.Nodes[i].ChildNodes.FindNode('line_name').NodeValue]));
        end;
      end;
    end;
  except
  end;
end;
```

■ Webサービスを利用するには

• SimpleAPI最寄駅APIを利用するには

Googleマップから取得した緯度・経度を使って、最寄駅を検索してみましょう。

```
FLocation.A
XMLDocume
XMLDocume
end;
end;
end else
begin
  ComboBox2.Items
  FLocation.Add(
end;
finally
  ComboBox2.ItemInd
  XMLDocument1.Acti
end;
end;
```

緯度 35.66271265557856
経度 139.73858686381527
最寄駅 六本木一丁目(323)
<最寄駅未選択>
六本木一丁目(323m)【東京メトロ南北線】

■ Webサービスを利用するには

• Google Geocoding APIを利用するには

Google Geocoding APIによる逆ジオコーディングについて

URL: <http://code.google.com/intl/ja/apis/maps>

[/documentation/geocoding/#ReverseGeocoding](http://code.google.com/intl/ja/apis/maps/documentation/geocoding/#ReverseGeocoding)

The screenshot shows the Google Maps API Web Services page. On the left, there is a navigation menu with links for 'Maps API ファミリー', 'KML サイトマップを作成', 'Maps API Web サービス', 'Geocoding API', 'Directions API', 'Elevation API', 'Places API', 'Maps JavaScript API V3', 'ホームページ', 'ドキュメント', 'Maps JavaScript API V2', and '(サポート終了予定の API)'. The main content area is titled 'Google Geocoding API' and contains several sub-sections: 'ジオコーディングとは', '対象読者', '使用制限', 'ジオコーディング リクエスト', 'ジオコーディング レスポンス', 'JSON 出力形式', 'XML 出力形式', 'ステータス コード', '結果', '住所コンポーネントのタイプ', '逆ジオコーディング', 'ビューポートのバイアス', and '地域のバイアス'. On the right, there is a detailed section titled '逆ジオコーディング(住所の逆検索)'. This section explains that geocoding is the process of converting a location on a map to a human-readable address. It states that the Geocoding API supports direct reverse geocoding using the `latlng` parameter. An example query is provided: `http://maps.google.com/maps/api/geocode/json?latlng=40.714224,-73.961452&sensor=true_or_false`. A note indicates that a space should not be used between the latitude and longitude values in the `latlng` parameter. Below the note, it says 'このクエリは、次の結果を返します:' followed by a JSON response:

```
{
  "status": "OK",
  "results": [ {
    "types": [ "street_address" ],
    "formatted_address": "275-291 Bedford Ave, Brooklyn, NY 11211, USA",
    "address_components": [ {
      "long_name": "275-291",
      "short_name": "275-291",
```



構造 ホームページ Unit1

- Panel1
 - CheckBox1
 - CheckBox2
 - ComboBox1
 - ComboBox2
 - Edit1
 - Edit2
 - Edit3
 - Edit4
 - Label1
 - Label2
 - RadioGroup1
 - SpeedButton1

オブジェクトインスペクタ

Edit4 TEdit

プロパティ イベント

| | |
|--------------------|--|
| ParentColor | <input type="checkbox"/> False |
| ParentCtl3D | <input checked="" type="checkbox"/> True |
| ParentCustomHint | <input checked="" type="checkbox"/> True |
| ParentDoubleBuffer | <input checked="" type="checkbox"/> True |
| ParentFont | <input checked="" type="checkbox"/> True |
| ParentShowHint | <input checked="" type="checkbox"/> True |
| PasswordChar | #0 |
| PopupMenu | |
| ReadOnly | <input type="checkbox"/> False |
| ShowHint | <input type="checkbox"/> False |
| TabOrder | 8 |
| TabStop | <input checked="" type="checkbox"/> True |
| Tag | 0 |
| Text | |
| TextHint | |

Text

Form1

マップタイプ

道路地図 混合表示

航空写真 地形情報

Zoom In Zoom Out

検索

マーカー 情報ウィンドウ

標準

緯度

経度

IdHTTP1 最寄駅

XML 住所

XMLDocument1

Project1.dproj - プロジェク...



ビルド構成

- ProjectGroup1
 - Project1.exe
 - ビルド構成
 - GoogleMap.pas
 - Unit1.pas

Unit1.pas

標準

緯度

経度

IdHTTP1 最寄駅

XML 住所

XMLDocument1

Proje... モデ... デー...

ツールパレット

- Standard
- Additional
- Win32
- System
- Win 3.1
- Dialogs
- Data Access
- Data Controls
- dbExpress
- Datasnap Server
- BDE
- Rave
- Vista Dialogs
- Gestures

■ Webサービスを利用するには

● Google Geocoding APIを利用するには

Googleマップから取得した緯度・経度を使って、住所を逆検索してみましょう。

The screenshot shows a web application window titled "Form1". On the left, there is a code editor with the following code:

```
procedure TForm1.SpeedButt
const
  cRequestURL = 'http://ma
begin
  // 住所問合せ
  XMLDocument1.XML.Text :=
  XMLDocument1.Active := T
  try
    if XMLDocument1.Docume
    begin
      Edit4.Text := XMLDoc
    end
    else
    begin
      Edit4.Text := '<住所
    end;
  finally
    XMLDocument1.Active :=
  end;
end;
```

The main area displays a Google Map of Tokyo. A red pin is placed on the map near the intersection of Roppongi and Nishi-Shinjuku. The map shows various landmarks and streets, including "六本木一丁目" (Roppongi 1-chome) and "神谷町" (Kamigaya-chome). The map type is set to "道路地図" (Road Map). The search bar on the right contains the text "日本, 東京都港区麻布台" (Japan, Tokyo, Minato-ku, Roppongi-dai).

On the right side of the map, there is a control panel with the following options:

- マップタイプ (Map Type):
 - 道路地図 (Road Map)
 - 混合表示 (Hybrid)
 - 航空写真 (Satellite)
 - 地形情報 (Terrain)
- Zoom In / Zoom Out buttons
- 検索 (Search) button
- マーカー (Marker)
- 情報ウィンドウ (Info Window)
- 標準 (Standard) dropdown menu
- 緯度 (Latitude): 35.66271265557856
- 経度 (Longitude): 139.73858686381527
- 最寄駅 (Nearest Station) dropdown menu
- 住所 (Address): 日本, 東京都港区麻布台

■ Googleマップを使ったアプリケーション例

● 取引先住所確認サンプル

The screenshot shows a web application window titled "MTS2011 Sample Form". It contains a table with customer information, address input fields, and a Google Maps interface.

| 取引先CD | 事業所CD | 取引先名 | 事業所名 | 郵便番号 | 住所1 | 住 |
|---------|-------|----------|-------|---------|-------------------|---|
| 10000 | 1 | 株式会社ミガロ. | 東京事業所 | 1060041 | 東京都港区麻布台1-4-3 | 工 |
| ▶ 10000 | 2 | 株式会社ミガロ. | 大阪本社 | 5560017 | 大阪府大阪市浪速区湊町2-1-57 | 業 |

緯度: 34.6651603 経度: 135.495413800000 位置保存

大阪府大阪市浪速区湊町2丁目1-57 最寄駅検索

JR難波(143m)【JR関西本線(大和路)】

<最寄駅未選択>

- JR難波(143m)【JR関西本線(大和路線)(加茂~JR難波)】
- なんば(239m)【大阪市営地下鉄四つ橋線】
- 大阪難波(405m)【阪神なんば線】
- 大阪難波(405m)【近鉄難波線】
- なんば(446m)【大阪市営地下鉄千日前線】
- なんば(463m)【大阪市営地下鉄御堂筋線】